IC 8696

Bureau of Mines Information Circular/1975

A Fortran Routine Reorganizer

UNITED STATES DEPARTMENT OF THE INTERIOR

Information Circular 8696

A Fortran Routine Reorganizer

By Marvin S. Seppanen
Twin Cities Mining Research Center, Twin Cities, Minn.

UNITED STATES DEPARTMENT OF THE INTERIOR
Thomas S. Kleppe, Secretary

Jack W. Carlson, Assistant Secretary–Energy and Minerals

BUREAU OF MINES
Thomas V. Falkie, Director

This publication has been cataloged as follows:

Seppanen, Marvin S

A Fortran routine reorganizer. [Washington] U.S. Bureau of Mines [1975]

62 p. illus., table. (U.S. Bureau of Mines. Information circular 8696)

Includes bibliography.

1. FORTRAN (Computer program language). 2. Mines and mineral resources—Computer programs. I. Title. (Series)

TN23.U71 no. 8696 622.06173

U.S. Dept. of the Int. Library

CONTENTS

ILLUSTRATIONS

TABLE

A FORTRAN ROUTINE REORGANIZER

by

Marvin S. Seppanen [1]

ABSTRACT

Computer programers are often required to make modifications to unfamiliar Fortran routines. This Bureau of Mines report describes a computer program designed to aid the programer in such a situation by reorganizing Fortran routines. This reorganization includes a sequential renumbering of the routine's statement numbers, a sequential renumbering and relocation of format statements, an alphanumeric reordering of dimensioned and typed variables, a uniform pattern of text spacing, and a sequential numbering of the records in the final Fortran routine.

The computer program has been extensively tested by the author and has proved to be a valuable tool for reorganizing Fortran routines developed under contract and later utilized by the Bureau, and for preparing routines for publication.

INTRODUCTION

Most computer programs are never finished to the programer's satisfaction because deadlines force the programer to leave the program at an intermediate working point short of the capabilities and options desired for the program. The program responsibility is often consigned to an operating agency, or the program is distributed to outside users, by the original programer. Even when the original programer remains in contact with his work, months can elapse before further work can be done to improve the logic. This transfer or delay means that most additional programing is done by programers unfamiliar, or out of date, with the coding. Working with an unfamiliar program is difficult because the logic is often scattered and the statement numbers are seldom in either a logical or numerical sequence. Also, during execution a program will, on occasion, terminate in an error condition, necessitating an error trace-back through the source program. This normally requires a study of the program's FORMAT statements and the associated output statements to determine where the error occurred in the logic. Finding a particular FORMAT statement in an unfamiliar program can be a difficult task.

---

[1] Operations research analyst (now with the University of Alabama, University, Ala.).

To aid the programer in these situations, a reorganization program (REOR) was developed by the Bureau of Mines to reorganize Fortran routines into a standard form. This reorganization includes a sequential renumbering of the routine's statement numbers, a sequential renumbering and relocation of format statements, and alphanumeric reordering of dimensioned and typed variables, a uniform pattern of text spacing, and a sequential numbering of the records in the final Fortran routine.

The reorganized Fortran routine follows the general conventions of programing style indicated by Kernighan and Plauger (3).[2] The resequencing of statement numbers is a major aid in avoiding unnecessary branches and in assuring that the routine's statement order follows the processing order. The uniform text spacing is beneficial when searching for potential error conditions. The indented DO loops provide a visual reminder to the programer to observe its limits.

Other programs such as TIDY are available that perform a function similar to REOR. REOR requires less computer memory than the University of Minnesota version of TIDY. That version of TIDY offers the user a large set of options not available to the REOR user. The REOR user is not required to individually specify those desirable options. REOR also does a more comprehensive reorganization of Hollerith fields in both FORMAT and DATA statements.

Being single purpose and written in a modular form, REOR can be easily modified for special Fortran conversion operations. For example, with a minor programing change REOR was used to identify special nonstandard mass storage input-output statements in one large set of programs being converted from one computer hardware to another. While not included in that case, REOR could be programed to automatically make such conversions.

THE PROGRAM (REOR)

The program (REOR) was written in the Fortran IV extended language for the Control Data Corp. (CDC) 6000 series computer[3] in the batch mode of operation. REOR reorganizes routines written in a code compatible with that computing system and other American National Standards Institute (ANSI) standard Fortran compilers. The reorganized routine code is general enough to allow its use with any ANSI standard Fortran compiler.

REOR uses one input file containing the original Fortran routines. It assumes that these routines are of a quality suitable for error-free compilation. The primary output is the file of reorganized Fortran routine statement. This file is in a form suitable for compilation, listing, or punching. REOR also prints information about its execution. A summary table is printed after each routine has been processed. REOR compensates for most errors in the routine being organized. When faced with an error condition, REOR makes the

---

[2]Underlined numbers in parentheses refer to items in the bibliography preceding the appendixes.

[3]Reference to specific equipment does not imply endorsement by the Bureau of Mines.

necessary assumptions to continue processing and prints messages to indicate the error condition and the actions taken. REOR checks each of its internal storage arrays to assure that they remain within bounds. Error messages indicate overflow conditions.

An effort has been made to make REOR's execution as efficient as possible; however, its execution does require a substantial amount of both central and peripheral processor time. Execution time is a function of the original routine's length and the individual statement's type, complexity, and length. Experience with the CDC 6600 has indicated that reorganizing a 100-record routine requires about 3.3 seconds of central processor time, and about 17 seconds of peripheral processor time. The CDC 6600 execution core requirement under the SCOPE operating system is 43,200 octal words.

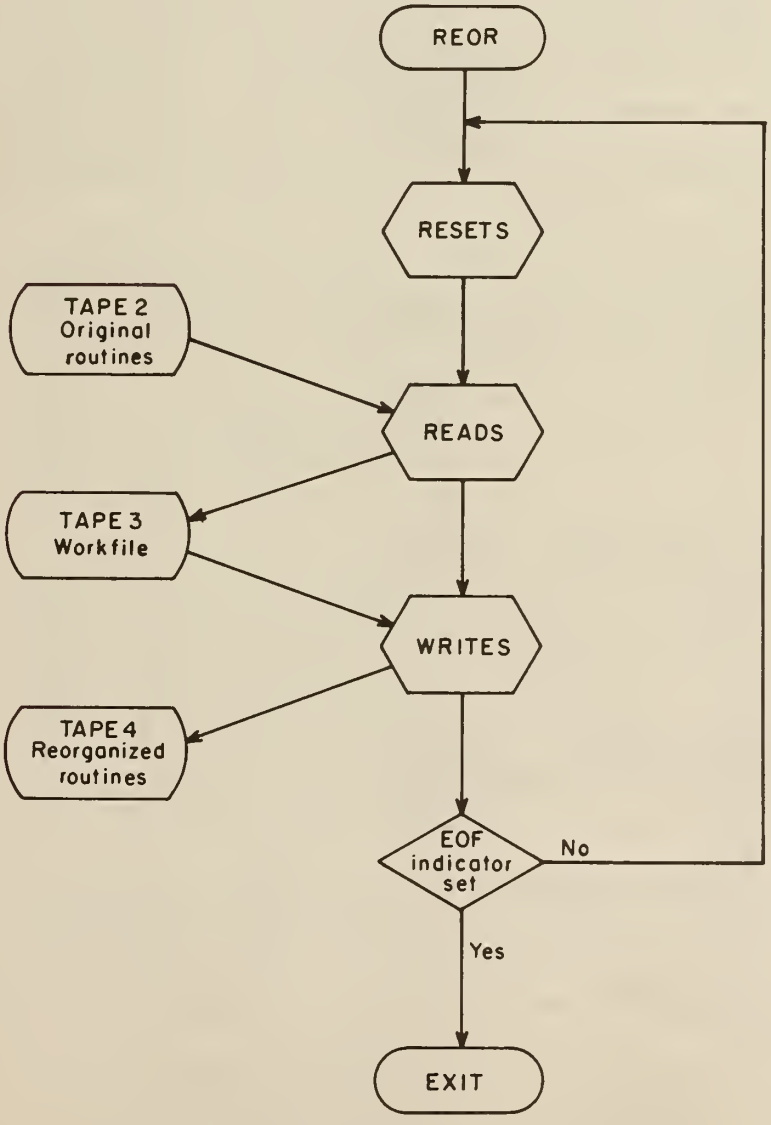

FIGURE 1. - REOR program macrologic.

REOR's output form is illustrated by its own list in appendix A. That list demonstrates most of REOR's capabilities. The individual REOR routines are documented with comment statements, and appendix B contains a functional description for each of the 27 separate routines. Appendix C is a list of definitions for the variables used by REOR. Appendix D illustrates the normal set of SCOPE control cards required to execute the program from an object code file REOR resident on disk.

REOR Logic

Program REOR operates through three basic processing cycles--read, write, and reset. Figure 1 illustrates the macro flow chart for the REOR program. The read cycle uses subroutine READS to read the original Fortran routine from logical unit TAPE 2, and to interpret each statement. The statements are either stored internally or written on a working file, TAPE 3. Figure 2 is a flow chart of the

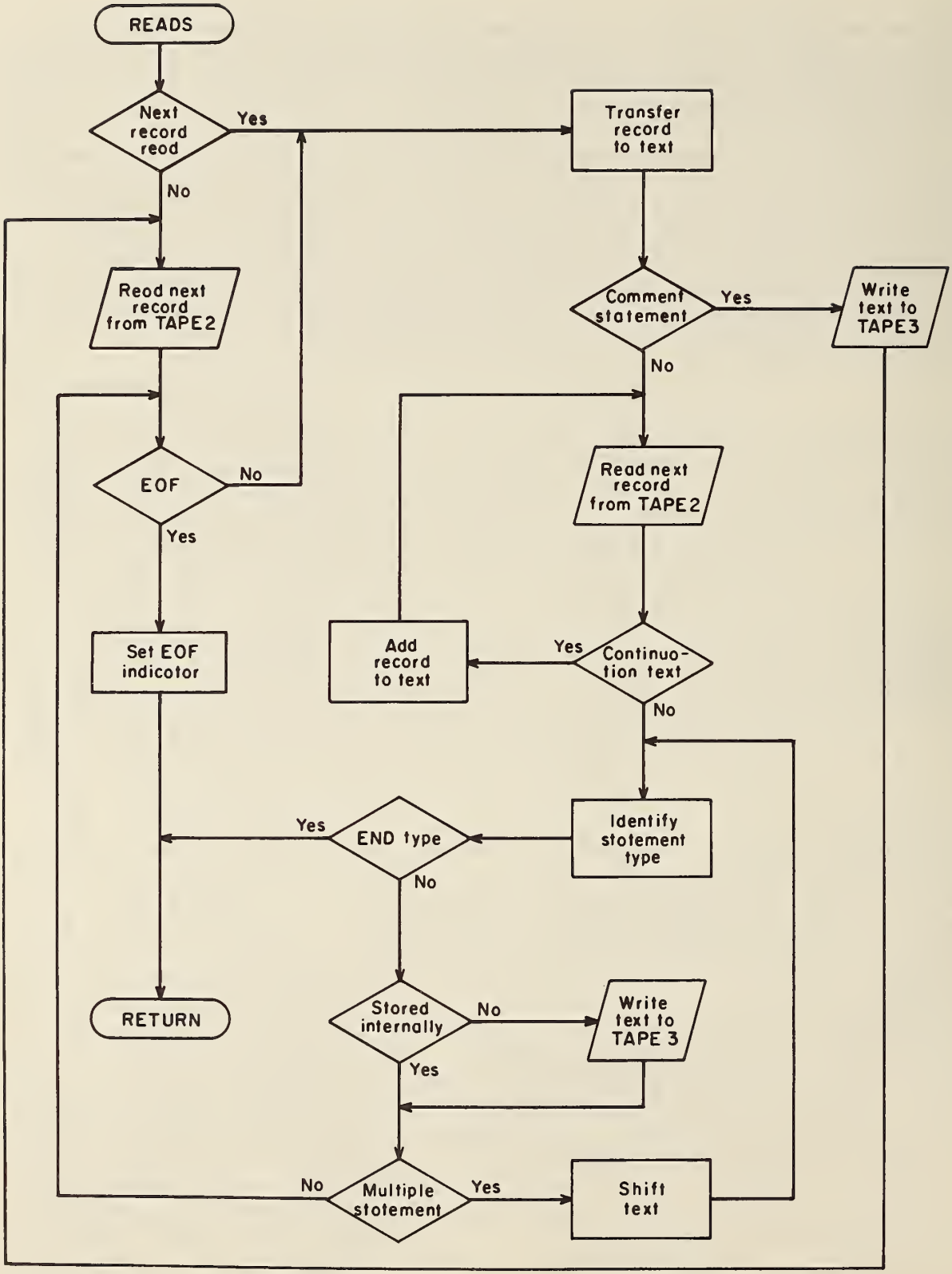

FIGURE 2. - READS subroutine logic.

READS subroutine logic. The write cycle uses subroutine WRITES to reconstruct and to write the reorganized Fortran routine on logical unit TAPE 4. Figure 3 is a flow chart of the WRITES subroutine logic. At the close of the write cycle REOR prints a summary of the processing for current routine. If the last reading operation from TAPE 2 encountered an End-of-File (EOF) mark, EXIT is called and the execution of REOR is terminated. If more routines are present on TAPE 2, subroutine RESETS is called to reset the counters and pointers. The read cycle is then repeated for the next routine. Each routine is processed independently.

Read Cycle

The read cycle individually processes each Fortran statement. These statements may occur on a single 80-column input record, or may be continued on one or more continuation records following the standard Fortran record layout convention--columns 1-5, statement number; column 6, continuation mark; columns 7-72; executable statements; and columns 73-80, comments. More than one executable statement may be contained in a single record if properly delineated with a dollar ($) sign. This feature agrees with CDC Fortran convention. To make the reorganized routine coding more widely compatible, all multiple statements are separated and processed individually.

When the read cycle is initiated, REOR first seeks to find the routine identification statement. This identification may be of one of the following types: PROGRAM, SUBROUTINE, FUNCTION, BLOCK DATA, or typed function. Failure to find such a statement results in an error condition. This condition is nonfatal, but does cause a message to be printed. The first four alphanumeric characters of the routine's name are retained for output labeling. If no valid identification statement is found, the label NAME is assumed. If the routine is of the BLOCK DATA type, it may not have a name, in which case the label DATA is assumed.

Each subsequent statement is identified and processed according to its characteristics. Variables defined by type of DIMENSION statements are collected, sorted in alphanumeric order, and stored internally for final processing. FORMAT statments are cataloged, condensed, and stored internally for final processing. Hollerith fields in the original routine can be of the following types: 4HTEXT, *TEXT*, or 'TEXT'. None of their internal spacing is altered. Consecutive *TEXT* or 'TEXT' type Hollerith fields separated by only spaces and/or a comma are combined into a single field by deleting the separating character and spaces.

With the exception of the END statement, the remaining statements are adjusted to obtain consistent internal spacing and written on the working file. The rules for internal spacing, except for Hollerith fields, follow:

1. Commas are always followed by a space.

2. Closed parentheses are preceded by a space.

3. Equal signs are preceded and followed by two spaces.

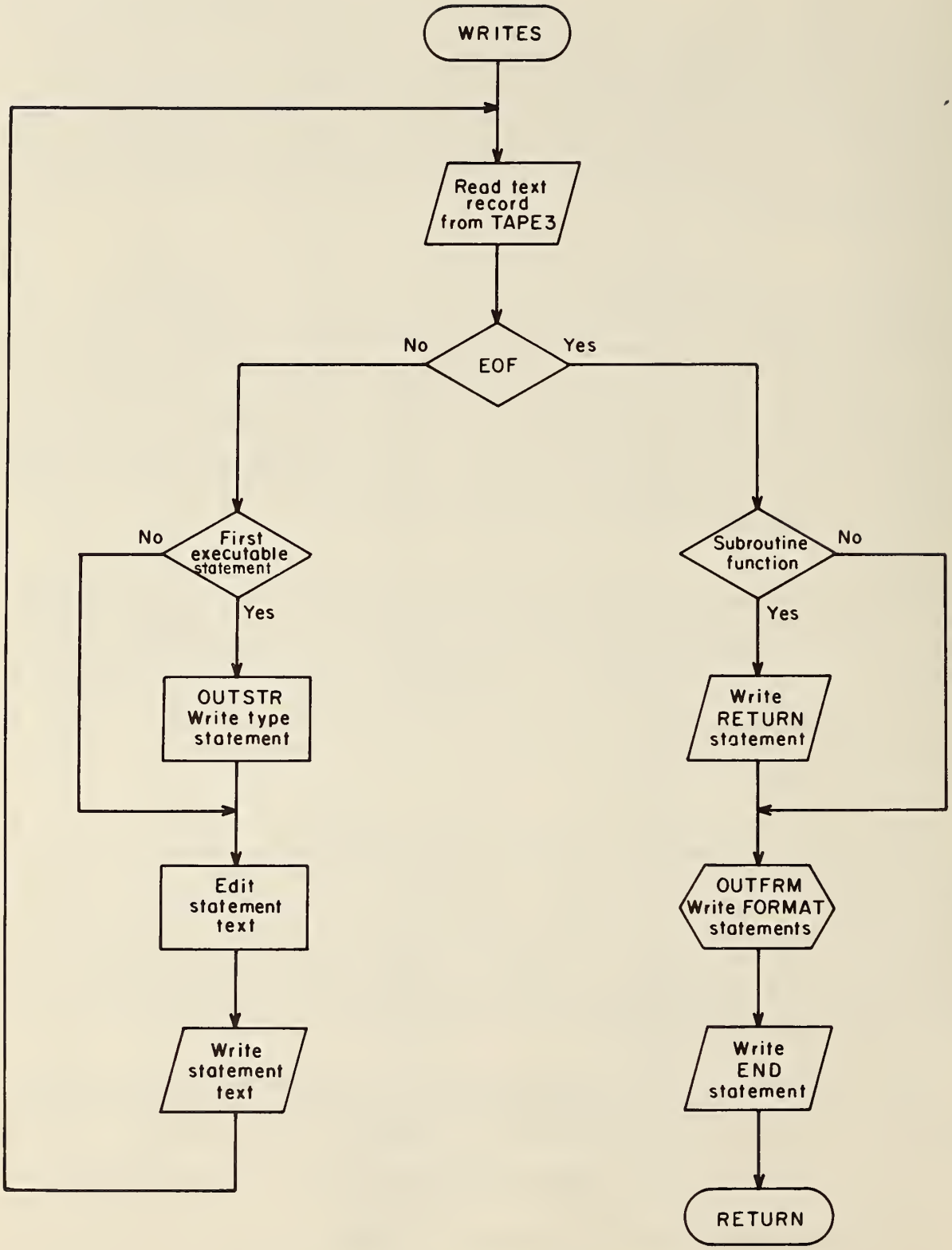

FIGURE 3. - WRITES subroutine logic.

4. Arithmetic symbols (+, *, -, or /) are preceded and followed by a space.

5. Logical operations within IF statements are preceded and followed by a space.

6. Special Fortran words (IF, DO, GO,...) are separated from other text by spaces.

Comment statement spacing is not altered. The letter C is inserted in column 1 of blank comment records or where the comment statement began with an asterisk. When more than one consecutive blank comment statement is found, all but the first are omitted. All Hollerith fields in DATA statements are converted to the 4HTEXT type. Statement types that internally contain statement numbers (GO TO 99, DO 99..., etc.) are scanned to locate those references. Statements whose type cannot be identified are assumed to be replacement expressions. These are scanned to assure that they contain an equal sign. Failure to find an equal sign is noted by an error message. Originally numbered executable statements are renumbered sequentially beginning with 1,000 in increments of 10. RETURN statements contained within a FUNCTION or SUBROUTINE are converted to GO TO 9999 statements. The last statement of the routine is of the RETURN type and, if necessary, is numbered 9999.

The read cycle is terminated when an END statement or an EOF is read. The working file is rewound, and the write cycle is started.

Write Cycle

The write cycle individually processes each routine statement on the working file. Each statement is composed into standard 80-column Fortran records. The text of each noncomment statement begins in column 8 of the output record. Statements extending beyond column 72 are continued on subsequent records. Whenever possible the text division for the continuation record is made at a space in the text. These continuation statements are numbered 1, 2, ..., 8, 9, ..., in column 6. Generally, the continuation record text is indented two spaces to the right from the first record. The exception to this rule occurs in FORMAT or DATA where 4HTEXT-type Hollerith fields continue from column 72 of one record onto column 7 of the next. To provide the indented form, *TEXT* and 'TEXT' fields in FORMAT statements are terminated in column 71 and resume in column 11 of the continuation record. The proper punctuation is inserted to retain the meaning of the text. DO loops are indented by an additional two columns to provide a readily noticed appearance. Concurrent DO loops are cumulatively indented two columns each. Each output record is labeled in columns 73-76 with the routine name or substitute generated in the read cycle. A sequence number beginning with and incremented by 10 is written in columns 77-80. Sequence numbers in excess of 9990 are prevented by shifting an asterisk to column 76 and restarting the sequence numbering with 10.

Prior to writing the first EQUIVALENCE, DATA, or executable statement, the dimensioned and typed variables are reconstructed from internal storage and written. All variables of a single type are placed in a single statement.

The order of the types used is DIMENSION, EXTERNAL, COMPLEX, DOUBLE, INTEGER, LOGICAL, and REAL. The redundant words PRECISION and TYPE are deleted from the output statements.

The executable statements are transferred from the working file to the output file. The new statement numbers assigned in the read cycle are used. Internal statement number references are changed to be consistent with the new numberings.

Prior to writing the END statement, any FORMAT statements are reconstructed from internal storage and written. The FORMAT statements are ordered according to their first use in the routine and are numbered sequentially 10, 20, 30, ..., 990. A referenced FORMAT statement that is not found in the original routine is noted with an error message and written as a default (A1) FORMAT statement in the reorganized routine. An unreferenced FORMAT statement is deleted.

Limitations

REOR is limited by the type of the Fortran code which can be used in the original routine, and by the size of the internal storage allocated for certain types of information. The language characteristics of several special CDC compilers have been incorporated into REOR. These include the special CDC input-output procedures BUFFERIN, BUFFEROUT, DECODE, and ENCODE. Other equipment-dependent functions have not been incorporated. REOR's internal information storage method places several limitations on the size of routines and statements it can process. These error conditions and recovery procedures are presented in table 1.

One Fortran condition is known to create an error situation. Because of its complex nature and infrequent use, it has not been handled by REOR. This condition arises from the CDC Fortran IV extended compiler capability to use Hollerith fields as arguments in routine calling statements, as logical operators, or as the right member of replacement expressions. Such usage will cause a potential problem when the Hollerith field contains an unequal number of left and right parentheses or blanks. This is a problem when the Hollerith field is part of the logical or arithmetic expression of an IF statement. Such a statement may not be properly handled, and an error message will note the condition.

REOR was coded for the CDC Fortran extended compiler. As such, it uses several CDC features that may not be available on all compilers. These include the EOF check function, the 10-character word length, and the DECODE and ENCODE statments. Use of these statements and function should be verified before attempting to compile REOR with other than the CDC Fortran extended compiler.

TABLE 1. - Error conditions and recovery procedures

Error message	Constraint variable	Recovery procedure	Detection location
Array STRING filled with more than 100 dimensioned or typed variables.	NMAX	This and all following typed or dimensioned variables deleted, processing continues.	STORE 440
Array LSTATE filled with more than 2,000 statement characters.	MLCHARS	Remainder of statement deleted, processing continues.	TRANSF 150
Array INNUM filled with more than 50 internal statement numbers.	NUMMAX	Remaining internal statement numbers not altered, processing continues.	KLIST 150
Array KFORM filled with more than 99 original FORMAT statement number calls.	MNFORM	Remaining original FORMAT statement number calls are not recorded, processing continues.	KF 220
Array KFOUT filled with more than 100 original FORMAT statement numbers	NFORM	Remaining FORMAT statements and numbers are not recorded, processing continues.	KO 220
Array LFOUT filled with more than 1,000 FORMAT statement words.	MFORM	Current FORMAT statement is not recorded, processing continues.	KO 270
Array KSNUM filled with more than 400 executable statements numbers.	MNSTATE	Routine deleted.....	READS 3380

BIBLIOGRAPHY

1. Control Data Corp. Control Data CYBER 70 Series Models 72/73/74 6000 Series Computer Systems, KRONOS 2.1 Time-Sharing Fortran Reference Manual. Pub. 60408600, Revision A, 1973, 140 pp.

2. _____. Control Data 6000 Computer Systems. 7600 Computer System, Fortran Extended Reference Manual, 6000 Version 3, 7600 Version 1. Pub. 60329100, Revision D, 1973, 225 pp.

3. Kernighan, B. W., and P. J. Plauger. The Elements of Programming Style. McGraw-Hill Book Co., Inc., New York, 1974, 147 pp.

4. McCracken, D. D. A Guide to Fortran IV Programming. John Wiley & Sons, Inc., New York, 2d ed., 1972, 288 pp.

5. University of Minnesota TIDY Documentation. Available in a computer list form from the University Computer Center, Minneapolis, Minn.

APPENDIX A.--PROGRAM LIST (REOR)

```
 PROGRAM     REOR                          CDC 6600 FTN V3.0-P355 OPT=1   06/25/75  12.55.39.

                      PROGRAM  REOR (TAPE2, TAPE3, TAPE4,  OUTPUT)                      REOR  10
                C     THIS PROGRAM READS A STANDARD FORTRAN ROUTINE FILE AND REORGANIZES REOR  20
                C       THE ROUTINE BY ORDERING THE STATEMENT NUMBERS AND ADJUSTING THE  REOR  30
                C       STATEMENT SPACING AND ORDER.                                    REOR  40
 5                      COMMON  /ALL/  1CHARS, 1DOLLAR, IERROR, 1NNUM (2, 50), 1POINT,   REOR  50
                     1    IPROG, ISNUM, ITYPE, 19999, KFORM (100), KFOUT (3, 100), KSNUM REOR  60
                     2    (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),     REOR  70
                     3    LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  REOR  80
                     4    NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING REOR  90
10                   5    (2, 100)                                                       REOR 100
                       COMMON  /DATA/  C, END, H, 1BLANK, 1EOF, 1NTEGER (10), 1PUNCT    REOR 110
                     1    (11), 1COUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,     REOR 120
                     2    MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,     REOR 130
                     3    STAR, X                                                        REOR 140
15                     INTEGER      C, END, H, 1DATA (4613), PROGRAM, RETURN, STAR,      REOR 150
                     1    STRING, X                                                      REOR 160
                       EQUIVALENCE  (1CHARS, 1CDATA(1))                                 REOR 170
                       DATA     1COUNT, 1DATA / 8 * 0, 4613 * 0 /                       REOR 180
                       DATA     C, END, H, 1BLANK, 1EOF / 1HC, 3HEND, 1HH, 1H , 0 /     REOR 190
20                     DATA     INTEGER / 1H0, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8,  REOR 200
                     1    1H9  /                                                         REOR 210
                       DATA     1PUNCT / 1H/, 1H+,, 1H(, 1H), 1H*, 1H$, 1H.,, 1H=, 1H-,  REOR 220
                     1    1H+ , 1H#  /                                                   REOR 230
                       DATA     LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS, MNFORM, MNSTATE,  REOR 240
25                   1    NCARD, NMAX, NUMMAX, RETURN, STAR, X / 2, 4,  3, 1000, 2000,   REOR 250
                     2    99, 400, 0, 100, 50, 6HRETURN, 1H*, 1HX  /                     REOR 260
                       DATA     PROGRAM / 1HP, 1HR, 1HO, 1HG, 1HR, 1HA, 1HM  /          REOR 270
                C     DO THE HOUSEKEEPING OPERATIONS.                                    REOR 280
                 1000 CALL RESETS                                                        REOR 290
30              C     DO THE READ CYCLE.  READ THE STATEMENTS FOR A ROUTINE FROM THE     REOR 300
                C       INPUT FILE TAPE2, PROCESS, AND STORE ON THE WORKING FILE TAPE10. REOR 310
                       CALL READS                                                       REOR 320
                C     DO THE WRITE CYCLE.  READ THE STATEMENTS FROM THE WORKING FILE,    REOR 330
                C       COMPLETE THE PROCESSING, AND WRITE ON THE OUTPUT FILE TAPE4.    REOR 340
35                     CALL WRITES                                                      REOR 350
                C     REPEAT IF NO EOF ENCOUNTERED.                                    REOR 360
                       IF (IEOF .EQ. 0)  GO TO 1000                                     REOR 370
                       CALL EXIT                                                        REOR 380
                       END                                                              REOR 390
```

SUBROUTINE BLANKS CDC 6600 FTN V3.0-P355 OPT=1 06/25/75 12.55.39.

```
      SUBROUTINE  BLANKS                                                BLAN  10
C     THIS ROUTINE SURPRESSES ALL BLANKS IN THE ARRAY LIST EXCEPT FOR   BLAN  20
C       THOSE IN HOLLERITH TYPE STATEMENTS.                             BLAN  30
      COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,    BLAN  40
     1   IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM BLAN  50
     2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),     BLAN  60
     3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  BLAN  70
     4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING BLAN  80
     5   (2, 100)                                                       BLAN  90
      COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT     BLAN 100
     1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,     BLAN 110
     2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,     BLAN 120
     3   STAR, X                                                        BLAN 130
      DIMENSION   LIST (1)                                              BLAN 140
      INTEGER     C, H, STAR, X                                         BLAN 150
      EQUIVALENCE  (LIST(1), LSTATE(1)), (ISTOP, LCHARS)                BLAN 160
      I  =  1                                                           BLAN 170
      IDOLLAR  =  0                                                     BLAN 180
      ISTOP  =  NONR (IBLANK, 1, ISTOP, LIST(1))                        BLAN 190
 1000 IF (SPRESS(1,ISTOP,LIST(1)) .NE. 0.0)  GO TO 1190                 BLAN 200
C     CHECK FOR A LEADING PUNCTUATION MARK.                             BLAN 210
 1010   DO 1020  J  =  1, 6                                            BLAN 220
        IF (LIST(I) .EQ. IPUNCT(J))  GO TO 1030                         BLAN 230
 1020   CONTINUE                                                        BLAN 240
      J  =  11                                                          BLAN 250
      IF (LIST(I) .EQ. IPUNCT(J))  GO TO 1040                           BLAN 260
      I  =  I + 1                                                       BLAN 270
      GO TO 1000                                                        BLAN 280
C     CHECK FOR A DOLLAR SIGN, $, INDICATING A MULTIPLE STATEMENT RECORD.BLAN 290
 1030 IF (J .EQ. 6)  GO TO 1180                                         BLAN 300
C     CHECK FOR A * TO BEGIN A HOLLERITH FIELD.                         BLAN 310
      IF (J .NE. 5)  GO TO 1050                                         BLAN 320
 1040 IF (ITYPE .EQ. 17)  GO TO 1130                                    BLAN 330
      IF (ITYPE .EQ. 16)  GO TO 1150                                    BLAN 340
C                                                                       BLAN 350
 1050 I  =  I + 1                                                       BLAN 360
 1060 IF (SPRESS(1,ISTOP,LIST(I)) .NE. 0.0)  GO TO 1190                 BLAN 370
        DO 1070  J  =  1, 10                                            BLAN 380
        IF (LIST(I) .EQ. INTEGER(J))  GO TO 1080                        BLAN 390
 1070   CONTINUE                                                        BLAN 400
      GO TO 1010                                                        BLAN 410
 1080 N  =  J - 1                                                       BLAN 420
 1090 I  =  I + 1                                                       BLAN 430
      IF (SPRESS(1,ISTOP,LIST(I)) .NE. 0.0)  GO TO 1190                 BLAN 440
C     CHECK FOR AN H,  INDICATES A HOLLERITH FIELD.                     BLAN 450
      IF (LIST(I) .EQ. H)  GO TO 1120                                   BLAN 460
C     CHECK FOR AN *,  INDICATES A MUTIPLE DATA ASSIGNMENT.             BLAN 470
      IF (LIST(I) .EQ. STAR)  GO TO 1050                                BLAN 480
C     CHECK FOR MORE NUMBERS.                                           BLAN 490
        DO 1100  J  =  1, 10                                            BLAN 500
        IF (LIST(I) .EQ. INTEGER(J))  GO TO 1110                        BLAN 510
 1100   CONTINUE                                                        BLAN 520
      GO TO 1010                                                        BLAN 530
 1110 N  =  N * 10 + J - 1                                              BLAN 540
      GO TO 1090                                                        BLAN 550
```

```
  SUBROUTINE  BLANKS                          CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

            C    SKIP THE ENTIRE H FIELD OF LENGTH N.                                    BLAN 560
             1120  I  =  I + 1 + N                                                       BLAN 570
                   GO TO 1010                                                           BLAN 580
            C   HERE FOR J = 17, FORMAT STATEMENTS.                                     BLAN 590
60          C    SKIP THE CHARACTERS BETWEEN THE * OR ≠ SIGNS.                          BLAN 600
            C   DELETE ANY TRAILING  COMMA.                                            BLAN 610
             1130  I  =  ISCANL (IPUNCT(J), I + 1, ISTOP, LIST(1)) + 1                  BLAN 620
                   IF (SPRESS(I,ISTOP,LIST(1)) .NE. 0.0)  GO TO 1190                    BLAN 630
                   IF (LIST(I) .NE. IPUNCT(2))  GO TO 1140                              BLAN 640
65                 CALL SHIFTL (IBLANK, I, ISTOP, LIST(1))                              BLAN 650
                   IF (SPRESS(I,ISTOP,LIST(1)) .NE. 0.0)  GO TO 1190                    BLAN 660
            C   COMBINE CONSECUTIVE SIMILIAR * OR ≠ FIELDS.                            BLAN 670
             1140  IF (LIST(I) .NE. IPUNCT(J))  GO TO 1010                             BLAN 680
                   CALL SHIFTL (IBLANK, I, ISTOP, LIST(1))                              BLAN 690
70                 I  =  I - 1                                                          BLAN 700
                   CALL SHIFTL (IBLANK, I, ISTOP, LIST(1))                              BLAN 710
                   IF (I .GT. ISTOP)  GO TO 1190                                        BLAN 720
                   GO TO 1130                                                           BLAN 730
            C   HERE FOR J = 16, DATA   STATEMENTS.                                    BLAN 740
75          C    CONVERT A *XXX* OR ≠XXX≠ TO A 3HXXX (CDC)                              BLAN 750
             1150  LIST (I)  =  H                                                      BLAN 760
                   II  =  I + 2                                                         BLAN 770
                   N  =  1                                                              BLAN 780
             1160  IF (II .GT. ISTOP)  GO TO 1190                                     BLAN 790
80                 IF (LIST(II) .EQ. IPUNCT(J))  GO TO 1170                           BLAN 800
                   II  =  II + 1                                                        BLAN 810
                   N  =  N + 1                                                          BLAN 820
                   GO TO 1160                                                           BLAN 830
             1170  CALL SHIFTL (IBLANK, II, ISTOP, LIST(1))                           BLAN 840
85                 CALL INSERTN (N, I, ISTOP, LIST(1))                                BLAN 850
                   GO TO 1060                                                           BLAN 860
            C                                                                          BLAN 870
             1180  IDOLLAR  =  I                                                       BLAN 880
                   ICHARS  =  IDOLLAR - 1                                           BLAN 890
90                 GO TO 9999                                                           BLAN 900
             1190  ICHARS  =  LCHARS                                                   BLAN 910
             9999  RETURN                                                             BLAN 920
                   END                                                                  BLAN 930
```

FUNCTION CHECK CDC 6600 FTN V3.0-P355 OPT=1 06/25/75 12.55.39.

```
      LOGICALFUNCTION  CHECK (LOOK4, NN, ISTART, ISTOP, LIST, IPOINT)  CHEC  10
C     THIS FUNCION SCANS A DATA LIST FOR A SPECIFIC DATA STRING (LOOK4). CHEC  20
      DIMENSION   LIST (1), LOOKUP (10)                                 CHEC  30
      DECODE (10,   10, LOOK4)  LOOKUP                                  CHEC  40
5     J  =  ISTART - 1                                                  CHEC  50
        DO 1020  I  =  1, NN                                            CHEC  60
        J  =  J + 1                                                     CHEC  70
1000    IF (J .GT. ISTOP)  GO TO 1030                                   CHEC  80
        IF (LIST(J) .NE. IBLANK)  GO TO 1010                            CHEC  90
10      CALL SHIFTL (IBLANK, J, ISTOP, LIST(1))                         CHEC 100
        GO TO 1000                                                      CHEC 110
1010    IF (LOOKUP(1) .NE. LIST(J))  GO TO 1030                         CHEC 120
1020    CONTINUE                                                        CHEC 130
      CHECK  =  .TRUE.                                                  CHEC 140
15    IPOINT  =  J + 1                                                  CHEC 150
      GO TO 9999                                                        CHEC 160
1030  CHECK  =  .FALSE.                                                 CHEC 170
      IPOINT  =  ISTART                                                 CHEC 180
9999  RETURN                                                            CHEC 190
20    C                                                                 CHEC 200
10    FORMAT  ( 100A1 )                                                 CHEC 210
      C                                                                 CHEC 220
      END                                                               CHEC 230
```

```
SUBROUTINE  FIXDATA                          CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                SUBROUTINE  FIXDATA                                                  FIXD  10
              C     THIS ROUTINE ASSURE THAT THE HOLLERITH FIELDS IN DATA STATEMENTS  FIXD  20
              C       ARE PROPERLY HANDLED.                                          FIXD  30
                COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,       FIXD  40
     5        I    IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM  FIXD  50
              2    (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),      FIXD  60
              3    LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,   FIXD  70
              4    NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING  FIXD  80
              5    (2, 100)                                                        FIXD  90
    10          COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT      FIXD 100
              1    (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,      FIXD 110
              2    MNFORM, MNSTATE, NCARC, NMAX, NUMMAX, PROGRAM (7), RETURN,      FIXD 120
              3    STAR, X                                                         FIXD 130
              C     SCAN FOR THE THE H WHICH MAY BE THE START OF A HOLLERITH FIELD  FIXD 140
    15          INTEGER     H                                                      FIXD 150
                II  =  10                                                          FIXD 160
             1000  IH  =  ISCANL (H, I1, LCHARS, LSTATE(1))                        FIXD 170
                IF (IH .GE. LCHARS)  GO TO 9999                                    FIXD 180
                IS  =  IH - 1                                                      FIXD 190
    20        C     DETERMINE IF THE H IS PRECEEDED BY AN INTEGER.                 FIXD 200
                IF (LSTATE(IS) .EQ. IBLANK)  GO TO 1080                            FIXD 210
                  DO 1010  I  =  1, 10                                             FIXD 220
                  IF (LSTATE(IS) .EQ. INTEGER(I))  GO TO 1020                      FIXD 230
             1010   CONTINUE                                                       FIXD 240
    25          GO TO 1080                                                         FIXD 250
             1020  N  =  I - 1                                                     FIXD 260
                IS  =  IS - 1                                                      FIXD 270
                IF (LSTATE(IS) .EQ. IBLANK)  GO TO 1070                            FIXD 280
                  DO 1030  I  =  1, 10                                             FIXD 290
    30            IF (LSTATE(IS) .EQ. INTEGER(I))  GO TO 1040                    FIXD 300
             1030   CONTINUE                                                       FIXD 310
                GO TO 1080                                                         FIXD 320
             1040  N  =  N + 10 * (1 - I)                                         FIXD 330
                IS  =  IS - I                                                      FIXD 340
    35          IF (LSTATE(IS) .EQ. IBLANK)  GO TO 1070                            FIXD 350
                  DO 1050  I  =  I, 10                                             FIXD 360
                  IF (LSTATE(IS) .EQ. INTEGER(I))  GO TO 1060                      FIXD 370
             1050   CONTINUE                                                       FIXD 380
                GO TO 1080                                                         FIXD 390
    40       1060  N  =  N + 100 * (I - 1)                                        FIXD 400
                IS  =  IS - 1                                                      FIXD 410
                IF (LSTATE(IS) .NE. IBLANK)  GO TO 1080                            FIXD 420
              C     DETERMINE IF THE INTEGER IS PRECEEDED BY A /, COMMA, OR *.     FIXD 430
             1070  IS  =  IS - 1                                                   FIXD 440
    45          IF (LSTATE(IS) .EQ. IPUNCT(1))  GO TO 1090                       FIXD 450
                IF (LSTATE(IS) .EQ. IPUNCT(2))  GO TO 1090                       FIXD 460
                IF (LSTATE(IS) .EQ. IPUNCT(5))  GO TO 1090                       FIXD 470
             1080  II  =  IH + 2                                                  FIXD 480
                GO TO 1000                                                         FIXD 490
    50       1090  IS  =  IH + N                                                  FIXD 500
                IH  =  IH + 1                                                      FIXD 510
                  DO 1100  I  =  IH, IS                                            FIXD 520
             1100   LSTATE (I)  =  LSTATE (I) + 1                                FIXD 530
                I1  =  IS + 3                                                      FIXD 540
    55          GO TO 1000                                                         FIXD 550
             9999  RETURN                                                         FIXD 560
                END                                                                FIXD 570
```

```
  FUNCTION    IDENT                          CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                FUNCTION  IDENT (N)                                                    IDEN  10
              C   THIS ROUTINE MATCHES CHARACTER STRINGS IN THE LIST ISTATE TO A        IDEN  20
              C     MASTER LIST, IA. WHERE,                                            IDEN  30
              C       IA (1,X)  IF THE CHARACTER IN THE LIST LSTATE EXCEEDS THE MATCH  IDEN  40
              C                   CHARACTER IN LA (2,X) THEN JUMP TO THIS POSITION.    IDEN  50
              C                   OTHERWISE EXIT, WITH IDENT = 45.                     IDEN  60
              C       IA (2,X)  THIS IS THE MATCH CHARACTER.                         IDEN  70
              C       IA (3,X)  WHEN A MATCH OCCURS THIS IS THE END CODE,            IDEN  80
              C                   = -  THIS MAY BE THE END OF THE STRING, HOWEVER,   IDEN  90
              C                          IT COULD CONTINUE TO A NEW VALUE.           IDEN 100
              C                          IF THE NEXT CHARACTER DOES NOT MATCH USE    IDEN 110
              C                          THE ABSOLUTE VALUE.                         IDEN 120
              C                   = 0  CONTINUE TO CHECK FOR FURTHER MATCHES         IDEN 130
              C                   = +  THIS IS THE END OF THE STRING USE THIS VALUE. IDEN 140
                COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,       IDEN 150
                1   IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM  IDEN 160
                2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),       IDEN 170
                3   LWORDS, NAME (4), NCARDS, XXXX, NFORMN, NFOUT, NKFORM, NOUTS,    IDEN 180
                4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING   IDEN 190
                5   (2, 100)                                                         IDEN 200
                COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT        IDEN 210
                1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,       IDEN 220
                2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,       IDEN 230
                3   STAR, X                                                          IDEN 240
                DIMENSION   IA (3,270), IB (3,78), IC (3,63), ID (3,47), IE          IDEN 250
                1   (3,70), IF (3,12), NNEXT (2)                                     IDEN 260
                EQUIVALENCE  (IA(1), IB(1))                                          IDEN 270
                EQUIVALENCE  (IA(235), IC(1))                                        IDEN 280
                EQUIVALENCE  (IA(424), ID(1))                                        IDEN 290
                EQUIVALENCE  (IA(565), IE(1))                                        IDEN 300
                EQUIVALENCE  (IA(775), IF(1))                                        IDEN 310
                DATA    IB / 9, 1HB, 0, 0, 1HL, 0, 0, 1HO, 0, 0, 1HC, 0, 0, 1HK,   IDEN 320
              1   0, 0, 1HD, 0, 0, 1HA, 0, 0, 1HT, 0, 0, 1HA, 4, 8, 1HC, 0, 0,     IDEN 330
              2   1HO, 0, 0, 1HM, 0, 0, 1HP, 0, 0, 1HL, 0, 0, 1HE, 0, 0, 1HX, 0,   IDEN 340
              3   16, 1H , 0, 15, 1HD, 0, 0, 1HO, 0, 0, 1HU, 0, 0, 1HB, 0, 0,      IDEN 350
              4   1HL, 0, 0, 1HE, 0, 0, 1HP, 0, 0, 1HR, 0, 0, 1HE, 0, 0, 1HC, 0,   IDEN 360
              5   0, 1HI, 0, 0, 1HS, 0, 0, 1HI, 0, 0, 1HO, 0, 0, 1HN, 0, 8, 1HF,   IDEN 370
              6   0, 0, 1HU, 0, 0, 1HN, 0, 0, 1HC, 0, 0, 1HT, 0, 0, 1HI, 0, 0,    IDEN 380
              7   1HO, 0, 0, 1HN, 3, 8, 1HI, 0, 0, 1HN, 0, 0, 1HT, 0, 0, 1HE, 0,   IDEN 390
              8   0, 1HG, 0, 0, 1HE, 0, 0, 1HR, 0,  - 15, 1H , 0, 8, 1HL, 0, 0,    IDEN 400
              9   1HO, 0, 0, 1HG, 0, 0, 1HI, 0, 0, 1HC, 0, 0, 1HA, 0, 0, 1HL, 0,   IDEN 410
              9   - 23, 1H , 0, 7, 1HP, 0, 0, 1HR, 0, 0, 1HO, 0, 0, 1HG, 0, 0,     IDEN 420
              9   1HR, 0, 0, 1HA, 0, 0, 1HM, 1, 5, 1HR, 0, 0, 1HE, 0, 0, 1HA, 0,   IDEN 430
              9   0, 1HL, 0,  - 35, 1H , 0, 0, 1HS, 0, 0, 1HU, 0, 0, 1HB, 0, 0,    IDEN 440
              9   1HR, 0, 0, 1HO, 0, 0, 1HU, 0, 0, 1HT, 0, 0, 1HI, 0, 0, 1HN, 0,   IDEN 450
              9   0, 1HE, 2 /                                                       IDEN 460
                DATA    IC / 6, 1HA, 0, 0, 1HS, 0, 0, 1HS, 0, 0, 1HI, 0, 0, 1HG,   IDEN 470
              1   0, 0, 1HN, 23, 21, 1HB, 0, 8, 1HA, 0, 0, 1HC, 0, 0, 1HK, 0, 0,  IDEN 480
              2   1HS, 0, 0, 1HP, 0, 0, 1HA, 0, 0, 1HC, 0, 0, 1HE, 40, 0, 1HU, 0,  IDEN 490
              3   0, 1HF, 0, 0, 1HF, 0, 0, 1HE, 0, 0, 1HR, 0, 3, 1HI, 0, 0, 1HN,   IDEN 500
              4   0, 0, 1H(, 30, 0, 1HO, 0, 0, 1HU, 0, 0, 1HT, 0, 0, 1H(, 31, 20,  IDEN 510
              5   1HC, 0, 3, 1HA, 0, 0, 1HL, 0, 0, 1HL, 22, 0, 1HO, 0, 9, 1HM, 0,  IDEN 520
              6   4, 1HM, 0, 0, 1HO, 0, 0, 1HN,  - 6, 0, 1H/, 5, 0, 1HP, 0, 0,     IDEN 530
              7   1HL, 0, 0, 1HE, 0, 0, 1HX, 9, 0, 1HN, 0, 0, 1HT, 0, 0, 1HI, 0,   IDEN 540
              8   0, 1HN, 0, 0, 1HU, 0, 0, 1HE, 24, 23, 1HD, 0, 3, 1HA, 0, 0,      IDEN 550
```

```
                9   1HT, 0, 0, 1HA, 16, 6, 1HE, 0, 0, 1HC, 0, 0, 1HO, 0, 0, 1HD, 0,IDEN 560
                9   0, 1HE, 0, 0, 1H(, 32, 8, 1HI, 0, 0, 1HM, 0, 0, 1HE, 0, 0, 1HN,IDEN 570
                9   0, 0, 1HS, 0, 0, 1HI, 0 /                                       IDEN 580
                  DATA    ID / 0, 1HO, 0, 0, 1HN, 7, 0, 1HO,  - 18, 0, 1HU, 0, 0,  IDEN 590
 60             1   1HB, 0, 0, 1HL, 0, 0, 1HE, 10, 33, 1HE, 0, 14, 1HN, 0, 5, 1HC, IDEN 600
                2   0, 0, 1HO, 0, 0, 1HD, 0, 0, 1HE, 0, 0, 1H(, 33, 5, 1HD,  - 44, IDEN 610
                3   0, 1HF, 0, 0, 1HI, 0, 0, 1HL, 0, 0, 1HE, 38, 0, 1HT, 0, 0, 1HR,IDEN 620
                4   0, 0, 1HY, 35, 1I, 1HG, 0, 0, 1HU, 0, 0, 1HI, 0, 0, 1HV, 0, 0, IDEN 630
                5   1HA, 0, 0, 1HL, 0, 0, 1HE, 0, 0, 1HN, 0, 0, 1HC, 0, 0, 1HE, 0, IDEN 640
 65             6   0, 1H(, 15, 0, 1HX, 0, 0, 1HT, 0, 0, 1HE, 0, 0, 1HR, 0, 0, 1HN,IDEN 650
                7   0, 0, 1HA, 0, 0, 1HL, 8, 7, 1HF, 0, 0, 1HO, 0, 0, 1HR, 0, 0,   IDEN 660
                8   1HM, 0, 0, 1HA, 0, 0, 1HT, 0, 0, 1H(, 17 /                     IDEN 670
                  DATA    IE / 5, 1HG, 0, 0, 1HO, 0, 0, 1HT, 0, 0, 1HO,  - 20, 0,  IDEN 680
                1   1H(, 19, 9, 1HI, 0, 2, 1HF, 0, 0, 1H(, 21, 0, 1HN, 0, 0, 1HT,  IDEN 690
 70             2   0, 0, 1HE, 0, 0, 1HG, 0, 0, 1HE, 0, 0, 1HR, 1I, 7, 1HL, 0, 0,  IDEN 700
                3   1HO, 0, 0, 1HG, 0, 0, 1HI, 0, 0, 1HC, 0, 0, 1HA, 0, 0, 1HL, 12,IDEN 710
                4   8, 1HN, 0, 0, 1HA, 0, 0, 1HM, 0, 0, 1HE, 0, 0, 1HL, 0, 0, 1HI, IDEN 720
                5   0, 0, 1HS, 0, 0, 1HT, 43, 20, 1HP, 0, 4, 1HA, 0, 0, 1HU, 0, 0, IDEN 730
                6   1HS, 0, 0, 1HE , 42, 11, 1HR, 0, 7, 1HE, 0, 0, 1HC, 0, 0, 1HI, IDEN 740
 75             7   0, 0, 1HS, 0, 0, 1HI, 0, 0, 1HO, 0, 0, 1HN, 14, 0, 1HI, 0, 0,  IDEN 750
                8   1HN, 0, 0, 1HT, 27, 0, 1HU, 0, 0, 1HN, 0, 0, 1HC, 0, 0, 1HH,   IDEN 760
                9   29, 14, 1HR, 0, 0, 1HE, 0, 4, 1HA, 0, 2, 1HD,  - 26, 0, 1H(,   IDEN 770
                9   25, 0, 1HL, 13, 4, 1HT, 0, 0, 1HU, 0, 0, 1HR, 0, 0, 1HN, 36, 0,IDEN 780
                9   1HW, 0, 0, 1HI, 0, 0, 1HN, 0, 0, 1HD, 39, 4, 1HS, 0, 0, 1HT, 0,IDEN 790
 80             9   0, 1HO, 0, 0, 1HP, 34, 5, 1HT, 0, 0, 1HY, 0, 0, 1HP, 0 /       IDEN 800
                  DATA    IF / 0, 1HE, 0,  - 154, 1H , 0, 4, 1HU, 0, 0, 1HS, 0, 0, IDEN 810
                1   1HE, 0, 0, 1H(, 37, 0, 1HW, 0, 0, 1HR, 0, 0, 1HI, 0, 0, 1HT, 0,IDEN 820
                2   0, 1HE, 0, 0, 1H(, 28 /                                         IDEN 830
                  DATA    NNEXT / 1, 79 /                                          IDEN 840
 85               ISTART  =  IPOINT                                                IDEN 850
                  NEXT  =  NNEXT (N)                                               IDEN 860
                  GO TO 1020                                                       IDEN 870
             1000 NEXT  =  NEXT + 1                                                IDEN 880
               C    ADVANCE TO THE CHARACTER OF THE LIST ISTATE.                   IDEN 890
 90              IPOINT  =  IPOINT + I                                            IDEN 900
             1010 IF (IPOINT .GT. ICHARS)  GO TO 1100                              IDEN 910
             1020 IF (LSTATE(IPOINT) .NE. IBLANK)  GO TO 1030                      IDEN 920
               C   SURPRESS ANY BLANKS.                                           IDEN 930
                  CALL SHIFTL (IBLANK, IPOINT, ICHARS, LSTATE(1))                  IDEN 940
 95               GO TO 1010                                                       IDEN 950
               C   NOW CHECK FOR A CHARACTER MATCH.                               IDEN 960
               C     IF ALREADY PAST USE THE DEFAULT TERMINATION, IDENT = 45.     IDEN 970
             1030 IF (LSTATE(IPOINT) .LT. IA(2,NEXT))  GO TO 1100                 IDEN 980
                  IF (LSTATE(IPOINT) .GT. IA(2,NEXT))  GO TO 1050                 IDEN 990
100            C   MATCH CONDITION.                                               IDENI000
               C    SEEK THE NEXT ACTION,                                         IDENI010
               C      = - SEARCH FOR POSSIBLE FURTHER ACTION.                     IDENI020
               C      = 0 CONTINUE.                                               IDENI030
               C      = + DONE.                                                   IDENI040
105            1040 IF (IA(3,NEXT))   1060, 1000, 1090                           IDENI050
               C    JUMP TO THE NEXT LEVEL  CHECK CHARACTER, DO NOT ADVANCE IPOINT. IDENI060
               C      = 0 DONE.                                                   IDENI070
               C      = + OR - JUMP TO THIS LOCATION IN IA(I,NEXT)  + NEXT.       IDENI080
             1050 IF (IA(1,NEXT) .EQ. 0)  GO TO 1100                             IDENI090
110                 NEXT  =  IA (1, NEXT) + NEXT                                 IDENI100
```

```
   FUNCTION    IDENT                                  CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                      GO TO 1030                                                          IDENI110
                C    IF NEGATIVE, THERE MAY BE ADDITIONAL CHARACTERS, IF NOT TAKE THIS VIDENI120
                C    ALUE OF IA (3,NEXT)                                                 IDENI130
                 1060 IDENT =   - IA (3, NEXT)                                           IDENI140
115             C    ADVANCE TO THE CHARACTER OF THE LIST ISTATE.                        IDENI150
                      IPOINT  =  IPOINT + 1                                              IDENI160
                 1070 IF (IPOINT .GT. ICHARS)  GO TO 9999                                IDENI170
                      IF (LSTATE(IPOINT) .NE. IBLANK)  GO TO 1080                        IDENI180
                C   SURPRESS ANY BLANKS.                                                 IDENI190
120                   CALL SHIFTL (IBLANK, IPOINT, ICHARS, LSTATE(I))                    IDENI200
                      GO TO 1070                                                         IDENI210
                 1080 NEXT  =  NEXT + 1                                                  IDENI220
                C   NOW CHECK FOR A CHARACTER MATCH.                                     IDENI230
                      IF (LSTATE(IPOINT) .EQ. IA(2,NEXT))  GO TO 1040                    IDENI240
125                   GO TO 9999                                                         IDENI250
                C    IF POSITIVE, WE ARE ALL DONE.                                      IDENI260
                 1090 IDENT  =  IA (3, NEXT)                                             IDENI270
                      IPOINT  =  IPOINT + 1                                              IDENI280
                      GO TO 9999                                                         IDENI290
130             C    A REPLACEMENT STATEMENT WAS APPARENTLY DETECTED.                    IDENI300
                 1100 IPOINT  =  ISTART                                                  IDENI310
                      IDENT  =  45                                                       IDENI320
                 9999 RETURN                                                             IDENI330
                      END                                                                IDENI340
```

```
SUBROUTINE  IFSPACE                          CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

      SUBROUTINE  IFSPACE                                                          IFSP  10
C                                                                                  IFSP  20
C   THIS ROUTINE COMPLETES THE SPACING WITHIN THE IF STATEMENTS.                   IFSP  30
C                                                                                  IFSP  40
      COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,               IFSP  50
     1   IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM           IFSP  60
     2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),                IFSP  70
     3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,             IFSP  80
     4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING            IFSP  90
     5   (2, 100)                                                                  IFSP 100
      COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT                IFSP 110
     1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,                IFSP 120
     2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,                IFSP 130
     3   STAR, X                                                                   IFSP 140
C                                                                                  IFSP 150
C   FIND THE FIRST ( - IPUNCT (3)                                                  IFSP 160
      LOGICAL     CHECK                                                            IFSP 170
      LLOWER  =  ISCANL (IPUNCT(3), 11, LCHARS, LSTATE(1))                         IFSP 180
C   FIND THE MATCHING ) - IPUNCT (4)                                               IFSP 190
      LUPPER  =  MATCH (LLOWER, LCHARS, LSTATE(1))                                 IFSP 200
C   FIND THE FIRST . - IPUNCT (7)                                                  IFSP 210
      IPFIRST  =  ISCANL (IPUNCT(7), LLOWER + 1, LCHARS, LSTATE(I))                IFSP 220
      IF (IPFIRST .GE. LUPPER)  GO TO 9999                                         IFSP 230
C   FIND THE NEXT  . - IPUNCT (7)                                                  IFSP 240
 1000 IPNEXT  =  ISCANL (IPUNCT(7), IPFIRST + 1, LCHARS, LSTATE(I))                IFSP 250
      IF (IPFIRST .GE. LUPPER)  GO TO 9999                                         IFSP 260
      IF (IPNEXT-IPFIRST .GT. 4)  GO TO 1080                                       IFSP 270
      IF (IPNEXT-IPFIRST-3)  1080, 1010, 1060                                      IFSP 280
C                                                                                  IFSP 290
C   2  CHARACTER SPACING. IS IT EQ, GE, GT, LE, LT, NE, OR.                        IFSP 300
 1010 IF (LSTATE(IPFIRST+1) .EQ. 1HE)  GO TO 1020                                  IFSP 310
      IF (LSTATE(IPFIRST+1) .EQ. 1HG)  GO TO 1030                                  IFSP 320
      IF (LSTATE(IPFIRST+1) .EQ. 1HL)  GO TO 1030                                  IFSP 330
      IF (LSTATE(IPFIRST+1) .EQ. 1HN)  GO TO 1040                                  IFSP 340
      IF (LSTATE(IPFIRST+1) .EQ. 1HO)  GO TO 1050                                  IFSP 350
      GO TO 1080                                                                   IFSP 360
C                                                                                  IFSP 370
 1020 IF (LSTATE(IPFIRST+2) .EQ. 1HQ)  GO TO 1070                                  IFSP 380
      GO TO 1080                                                                   IFSP 390
C                                                                                  IFSP 400
 1030 IF (LSTATE(IPFIRST+2) .EQ. 1HT)  GO TO 1070                                  IFSP 410
 1040 IF (LSTATE(IPFIRST+2) .EQ. 1HE)  GO TO 1070                                  IFSP 420
      GO TO 1080                                                                   IFSP 430
C                                                                                  IFSP 440
 1050 IF (LSTATE(IPFIRST+2) .EQ. 1HR)  GO TO 1070                                  IFSP 450
      GO TO 1080                                                                   IFSP 460
C                                                                                  IFSP 470
C   3  CHARACTER SPACING. IS IT AND OR NOT.                                        IFSP 480
 1060 IF (CHECK(3HAND,3,IPFIRST+1,IPNEXT,LSTATE(1),IP))  GO TO 1070                IFSP 490
      IF ( .NOT. CHECK(3HNOT,3,IPFIRST+1,IPNEXT,LSTATE(I),IP))  GO TO              IFSP 500
     1   1080                                                                      IFSP 510
C                                                                                  IFSP 520
C   YES, INSERT SURROUNDING SPACES.                                                IFSP 530
 1070 CALL INSERT (IBLANK, IPNEXT + 1, LCHARS, LSTATE(I), 1)                       IFSP 540
      CALL INSERT (IBLANK, IPFIRST, LCHARS, LSTATE(I), 1)                          IFSP 550
      IPNEXT  =  IPNEXT + 2                                                        IFSP 560
 1080 IPFIRST  =  IPNEXT                                                           IFSP 570
      GO TO 1000                                                                   IFSP 580
C                                                                                  IFSP 590
 9999 RETURN                                                                       IFSP 600
      END                                                                          IFSP 610
```

SUBROUTINE INSERT CDC 6600 FTN V3.0-P355 OPT=1 06/25/75 12.55.39.

```
      SUBROUTINE  INSERT (NEW, ISTART, ISTOP, LIST, N)                  INSE  10
C     THIS ROUTINE INSERTS INTO THE DATA STRING LIST THE N CHARACTERS PASINSE  20
C       THRU NEW, START AT POSITION ISTART.  ISTOP IS INCREASED BY N.   INSE  30
        COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,  INSE  40
     1    IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM INSE  50
     2    (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),     INSE  60
     3    LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  INSE  70
     4    NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING INSE  80
     5    (2, 100)                                                       INSE  90
        COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT    INSE  92
     1    (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,     INSE  94
     2    MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,     INSE  96
     3    STAR, X                                                        INSE  98
        DIMENSION   LIST (1), NEW (1), NEWTEMP (100)                    INSE 100
        NN  =  N                                                        INSE 110
        IF (NN .LE. 0)  GO TO 9999                                      INSE 120
        IF (NUMIN .LE. 0)  GO TO 1010                                   INSE 130
          DO 1000  J  =  1, NUMIN                                       INSE 140
          IF  (ISTART .LT. INNUM(1, J))  INNUM (1, J)  =  INNUM (1, J) + INSE 150
     1      NN                                                         INSE 160
 1000   CONTINUE                                                        INSE 170
 1010   DECODE (NN,   10, NEW (1))   (NEWTEMP (I), I=1, NN)            INSE 180
 1020     DO 1030  I  =  1, NN                                         INSE 190
 1030     CALL SHIFTR (NEWTEMP(I), ISTART + I - 1, ISTOP, LIST(1))     INSE 200
        ICHARS  =  ICHARS + NN                                          INSE 210
        IF  (IDOLLAR .GT. 0)  IDOLLAR  =  IDOLLAR + NN                  INSE 220
        GO TO 9999                                                      INSE 230
        ENTRY INSERTN                                                   INSE 240
        ENCODE (10,   20, NEWTEMP (100))  NEW (1)                       INSE 250
        DECODE (5,   10, NEWTEMP (100))   (NEWTEMP (I), I=1, 5)        INSE 260
        NN  =  5                                                        INSE 270
        IF (N .GE. 4)  GO TO  1020                                      INSE 272
 1040   IF (NEWTEMP (2) .NE. IBLANK)  GO TO 1020                        INSE 273
        NN  =  NN - 1                                                   INSE 276
        DO  1050  I = 2, NN                                             INSE 280
 1050   NEWTEMP (I)  =  NEWTEMP (I+ 1)                                 INSE 283
        GO TO  1040                                                     INSE 286
        ENTRY INSERTS                                                   INSE 290
        NN  =  N                                                        INSE 300
        IF (NN .LE. 0)  GO TO 9999                                      INSE 310
        GO TO 1010                                                      INSE 320
 9999   RETURN                                                          INSE 330
C                                                                       INSE 340
   10   FORMAT  ( 100A1 )                                              INSE 350
   20   FORMAT  ( I5 )                                                 INSE 360
C                                                                       INSE 370
        END                                                             INSE 380
```

```
  FUNCTION    ISCANR                                CDC 6600 FTN V3.0-P3SS OPT=I  06/25/7S  I2.SS.39.

                      FUNCTION  ISCANR (LOOK4, ISTART, ISTOP, LIST)                    ISCA  I0
                    C     THIS FUNCION SCANS A DATA LIST FOR A SPECIFIC CHARACTER   (LOOK4). ISCA  20
                    C       AND RETURNS THE LOCATION IF FOUND.                       ISCA  30
                    C     SCAN FROM THE RIGHT (LAST).                                ISCA  40
               S            DIMENSION   LIST (I)                                     ISCA  50
                            I  =  ISTOP                                              ISCA  60
                     I000   IF (I .LT. ISTART)  GO TO I020                           ISCA  70
                            IF (LIST(I) .EQ. LOOK4)  GO TO I020                      ISCA  80
                            I  =  I - I                                              ISCA  90
              I0            GO TO I000                                               ISCA I00
                    C                                                                ISCA I10
                            ENTRY ISCANL                                             ISCA I20
                    C     SCAN FROM THE LEFT (FIRST).                                ISCA I30
                            I  =  ISTART                                             ISCA I40
              IS     I0I0   IF (I .GT. ISTOP)  GO TO I020                           ISCA I50
                            IF (LIST(I) .EQ. LOOK4)  GO TO I020                      ISCA I60
                            I  =  I + I                                              ISCA I70
                            GO TO I0I0                                               ISCA I80
                     I020   ISCANR  =  I                                             ISCA I90
              20     9999   RETURN                                                   ISCA 200
                            END                                                      ISCA 2I0
```

```
SUBROUTINE  KF                                  CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                SUBROUTINE  KF (NSTN)                                                  KF    10
              C     THIS ROUTINE CATALOGS THE FORMAT STATEMENT NUMBER IN THE ORDER OF   KF    20
              C       THEIR USE IN THE ROUTINE.                                        KF    30
              C                                                                        KF    40
    5           COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,       KF    50
                1    IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM  KF    60
                2    (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),      KF    70
                3    LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,   KF    80
                4    NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING  KF    90
   10           5    (2, 100)                                                        KF   100
                COMMON  /DATA/  C, END, F, IBLANK, IEOF, INTEGER (10), IPUNCT       KF   110
                1    (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,      KF   120
                2    MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,      KF   130
                3    STAR, X                                                         KF   140
   15           IF (NFORMN .LE. 0)  GO TO 1010                                       KF   150
              C     CHECK IF THIS FORMAT STATEMENT NUMBER IS ALREADY CATELOGED.       KF   160
                   DO 1000  J  =  I, NFORMN                                        KF   170
                   IF (KFORM(J) .EQ. NSTN)  GO TO 1020                             KF   180
               1000    CONTINUE                                                    KF   190
   20         C     CATELOG AT THE END OF THE ARRAY.                                KF   200
               1010  NFORMN  =  NFORMN + 1                                        KF   210
                   IF (NFORMN .GT. MNFORM)  GO TO 1030                             KF   220
                   KFORM (NFORMN)  =  NSTN                                         KF   230
               1020  GO TO 9999                                                    KF   240
   25          1030  PRINT   10, MNFORM, (LSTATE(I), I=I, LCHARS)                  KF   250
               9999  RETURN                                                        KF   260
              C                                                                    KF   270
                 10  FORMAT  (  *0THE ARRAY (KFORM) IS FULL.  THE NUMBER OF FORMAT ST*KF   280
                   I   *ATEMENT NUMBERS CATALOGED BY THEIR ORDER OF FIRST USE EXCEEDE*KF   290
   30              2   *D * IS *ON STATEMENT* // (20X, I00A1) )                     KF   300
              C                                                                    KF   310
                   END                                                             KF   320
```

```
  FUNCTION    KLIST                                  CDC 6600 FTN V3.0-P355 OPT=1   06/25/75  12.55.39.

                LOGICALFUNCTION  KLIST (IP, NSTN)                                    KLIS  10
              C     THIS FUNCTION RECORDS THE VALUE AND THE POSITION OF THE INTERNAL   KLIS  20
              C       STATEMENT NUMBERS.                                             KLIS  30
                      COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,  KLIS  40
    5               1   IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM KLIS  50
                    2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),     KLIS  60
                    3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  KLIS  70
                    4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING KLIS  80
                    5   (2, 100)                                                     KLIS  90
   10               COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT    KLIS 100
                    1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,   KLIS 110
                    2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,   KLIS 120
                    3   STAR, X                                                      KLIS 130
                      KLIST  =  .FALSE.                                              KLIS 140
   15                 IF (NUMIN .GE. NUMMAX .OR. NUMIN .LT. 0)  GO TO 1000           KLIS 150
                      NUMIN  =  NUMIN + 1                                            KLIS 160
                      INNUM (1, NUMIN)  =  IP                                        KLIS 170
                      INNUM (2, NUMIN)  =  NSTN                                      KLIS 180
                      KLIST  =  .TRUE.                                               KLIS 190
   20                 GO TO 9999                                                     KLIS 200
                1000  PRINT   10, NUMMAX, (LSTATE(I), I=1, LCHARS)                   KLIS 210
                9999  RETURN                                                        KLIS 220
              C                                                                      KLIS 230
                  10  FORMAT  (  *0THE ARRAY (INNUM) IS FULL.  THE NUMBER OF INTERNAL *KLIS 240
   25               1   *STATEMENT NUMBERS EXCEEDED * I5 *ON STATEMENT* // (20X, 100A1)KLIS 250
                    2   )                                                            KLIS 260
              C                                                                      KLIS 270
                      END                                                            KLIS 280
```

```
FUNCTION    KO                                  CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                LOGICALFUNCTION  KO (NSTN)                                          KO    10
              C     THIS FUNCTION CATELOGS THE LOCATION OF THE FORMAT STATEMENT NUMBERSKO    20
              C       IN THE ARRAY KFOUT.                                           KO    30
                      COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,  KO    40
    5                 1   IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM KO    50
                      2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),    KO    60
                      3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  KO    70
                      4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING KO    80
                      5   (2, 100)                                                  KO    90
   10                 COMMON  /DATA/  C, ENO, H, IBLANK, IEOF, INTEGER (10), IPUNCT   KO   100
                      I   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,  KO   110
                      2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,  KO   120
                      3   STAR, X                                                   KO   130
                        KO  =  .FALSE.                                              KO   140
   15                 IF (NFOUT .LE. 0)  GO TO 1010                                KO   150
              C     CHECK IF THIS FORMAT STATEMENT NUMBER IS ALREADY CATELOGED.      KO   160
                      DO 1000  J  =  1, NFOUT                                       KO   170
                      IF (KFOUT(I,J) .EQ. NSTN)  GO TO 9999                         KO   180
               1000   CONTINUE                                                      KO   190
   20         C    CATELOG AT THE END OF THE ARRAY.                                KO   200
               1010   NFOUT  =  NFOUT + 1                                          KO   210
                      IF (NFOUT .GT. MNFORM)  GO TO 1020                            KO   220
                      KFOUT (1, NFOUT)  =  NSTN                                     KO   230
                      KFOUT (2, NFOUT)  =  NEXT                                     KO   240
   25                 KFOUT (3, NFOUT)  =  ICHARS                                  KO   250
                      NEXT  =  NEXT + (ICHARS + 9) / 10                             KO   260
                      IF (NEXT .GT. MFOUT)  GO TO 1030                              KO   270
                      KO  =  .TRUE.                                                 KO   280
                      GO TO 9999                                                    KO   290
   30                 1020   PRINT   10, MNFORM, (LSTATE(I), I=I, LCHARS)          KO   300
                      GO TO 9999                                                    KO   310
                      1030   PRINT   20, MFOUT, (LSTATE(I), I=I, LCHARS)          KO   320
                      NEXT  =  KFOUT (2, NFOUT)                                     KO   323
                      NFOUT  =  NFOUT - I                                           KO   326
   35                 9999   RETURN                                                KO   330
              C                                                                     KO   340
                 10   FORMAT  (  *0THE ARRAY (KFOUT) IS FULL.  THE NUMBER OF FORMAT ST*KO   350
                   1    *ATEMENT NUMBER STORED IN ARRAY (KSNUM) EXCEEDED * I5       KO   360
                   2    *ON STATEMENT* // (20X, I00A1) )                            KO   370
   40              20   FORMAT  (  *0THE ARRAY (LFOUT) IS FULL.  THE NUMBER OF FORMAT ST*KO   380
                   1    *ATEMENT   WORDS    EXCEEDED * I5 *ON STATEMENT* // (20X,    KO   390
                   2    100A1) )                                                    KO   400
              C                                                                     KO   410
                      END                                                           KO   420
```

```
 FUNCTION    MATCH                                CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                FUNCTION  MATCH (ISTART, ISTOP, LIST)                                    MATC  10
              C     THIS FUNCTION FINDS THE CLOSING ).                                   MATC  20
              C       ISTART      KNOWN POSITION OF THE FIRST (.                         MATC  30
                      COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT      MATC  40
    5                 1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,    MATC  50
                      2   MNFORM, MNSTATE, NCARC, NMAX, NUMMAX, PROGRAM (7), RETURN,     MATC  60
                      3   STAR, X                                                        MATC  70
                        DIMENSION   LIST (1)                                             MATC  80
                        I4  =  I3  =  ISTART + 1                                         MATC  90
   10               C       I3    POSITION OF NEXT (.                                   MATC 100
                     1000   I3  =  ISCANL (IPUNCT(3), I3, ISTOP, LIST(I))                MATC 110
                    C       I4    POSITION OF NEXT ).                                   MATC 120
                            I4  =  MATCH  =  ISCANL (IPUNCT(4), I4, ISTOP, LIST(1))      MATC 130
                    C     LAST ) IS FOUND WHEN NEXT ( IS TO THE RIGHT OR WHEN ISTOP HAS BEEN MATC 140
   15               C       EXCEEDED.                                                     MATC 150
                            IF (I3 .GE. I4 .OR. I4 .GT. ISTOP)  GO TO 9999               MATC 160
                    C     ACCROSS BY PAIRS.                                              MATC 170
                            I3  =  I3 + 1                                                MATC 180
                            I4  =  I4 + 1                                                MATC 190
   20                       GO TO 1000                                                   MATC 200
                     9999   RETURN                                                       MATC 210
                            END                                                          MATC 220
```

```
  FUNCTION    NONR                                CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                      FUNCTION  NONR (LOOK4, ISTART, ISTOP, LIST)                      NONR  10
                C     THIS FUNCION DETERMINES THE LOCATION OF THE LAST (NONR) OR FIRST  NONR  20
                C       (NONL) NONE (LOOK4)                                            NONR  30
                C       CHARACTER BETWEEN ISTART AND ISTOP IN THE DATA STRING LIST.     NONR  40
 5              C     SCAN FROM THE RIGHT (LAST).                                      NONR  50
                      DIMENSION   LIST (1)                                             NONR  60
                      I  =  ISTOP                                                      NONR  70
                 1000 IF (I .LT. ISTART)  GO TO 1020                                   NONR  80
                      IF (LIST(I) .NE. LOOK4)  GO TO 1020                              NONR  90
10                    I  =  I - 1                                                      NONR 100
                      GO TO 1000                                                       NONR 110
                C                                                                      NONR 120
                      ENTRY NONL                                                       NONR 130
                C     SCAN FROM THE LEFT (FIRST).                                      NONR 140
15                    I  =  ISTART                                                     NONR 150
                 1010 IF (I .GT. ISTOP)  GO TO 1020                                    NONR 160
                      IF (LIST(I) .NE. LOOK4)  GO TO 1020                              NONR 170
                      I  =  I + 1                                                      NONR 180
                      GO TO 1010                                                       NONR 190
20               1020 NONR  =  I                                                       NONR 200
                 9999 RETURN                                                           NONR 210
                      END                                                              NONR 220
```

```
FUNCTION    NUMBS                                CDC 6600 FTN V3.0-P355 OPT=I   06/25/75  12.55.39.

                FUNCTION  NUMBS (ISTART, ISTOP, LIST)                                NUMB  10
              C     THIS FUNCTION EXAMINES THE STRING LIST STARTING AT ISTART LOOKING  NUMB  20
              C       FOR A NUMERICAL VALUE WHICH IF FOUND IS RETURNED AND THE LOCATIONNUMB  30
              C       SURPRESSED, OTHERWISE A ZERO IS RETURNED.                      NUMB  40
 5                    COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,   NUMB  50
                    I   IPROG, ISNUM, ITYPE, I9999, KFORM (I00), KFOUT (3, I00), KSNUM NUMB  60
                    2   (2, 400), LCARD (80), LCHARS, LFOUT (I000), LSTATE (2000),     NUMB  70
                    3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  NUMB  80
                    4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING NUMB  90
10                  5   (2, I00)                                                       NUMB I00
                    COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (I0), IPUNCT      NUMB I10
                    1   (II), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,    NUMB I20
                    2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,    NUMB I30
                    3   STAR, X                                                        NUMB I40
15                  DIMENSION   LIST (1)                                               NUMB I50
                    IS  =  ISTART                                                      NUMB I60
                    NUMBS  =  0                                                        NUMB I70
               1000 IF (IS .GT. ISTOP)  GO TO 9999                                     NUMB I80
                    IF (LIST(IS) .EQ. IBLANK)  GO TO I030                              NUMB I90
20                    DO I0I0  I  =  I, 10                                             NUMB 200
                      IF (LIST(IS) .EQ. INTEGER(I))  GO TO 1020                        NUMB 210
               I0I0   CONTINUE                                                         NUMB 220
                    GO TO 9999                                                         NUMB 230
               I020 NUMBS  =  NUMBS * I0 + I - I                                       NUMB 240
25             I030 CONTINUE                                                           NUMB 250
                    CALL SHIFTL (IBLANK, IS, ISTOP, LIST(1))                           NUMB 260
                    ICHARS  =  ICHARS - I                                              NUMB 270
                    IF  (IDOLLAR .GT. 0)  IDOLLAR  =  IDOLLAR - I                      NUMB 280
                    GO TO I000                                                         NUMB 290
30             9999 RETURN                                                             NUMB 300
                    END                                                                NUMB 310
```

```
SUBROUTINE  OUTFRM                                  CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

      SUBROUTINE  OUTFRM                                                        OUTF  10
C     THIS ROUTINE OUTPUTS THE FORMAT STATEMENTS IN THE ORDER THEY ARE          OUTF  20
C       USED.                                                                   OUTF  30
        COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,          OUTF  40
     1    IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM       OUTF  50
     2    (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),           OUTF  60
     3    LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,        OUTF  70
     4    NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING       OUTF  80
     5    (2, 100)                                                             OUTF  90
        COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT           OUTF 100
     1    (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,           OUTF 110
     2    MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,           OUTF 120
     3    STAR, X                                                              OUTF 130
        INTEGER     A1, C, FORMAT, H                                            OUTF 140
        DATA    A1 / 2HA1  / , FORMAT / 6HFORMAT  /                             OUTF 150
        IF (NFORMN .LE. 0 .OR. IERROR .EQ. 999)  GO TO 9999                     OUTF 160
        LCHARS  =  1                                                            OUTF 170
        LSTATE (1)  =  C                                                        OUTF 180
        CALL PUNCHIT (0)                                                        OUTF 190
        IERROR  =  999                                                          OUTF 200
          DO 1240  J  =  1, NFORMN                                              OUTF 210
            DO 1000  JJ  =  1, NFOUT                                            OUTF 220
            IF (KFORM(J) .EQ. KFOUT(1,JJ))  GO TO 1010                          OUTF 230
 1000       CONTINUE                                                           OUTF 240
C     NOT FOUND INSERT THE DUMMY FORMAT STATEMENT.                              OUTF 250
          PRINT   10, KFORM (J)                                                 OUTF 260
          LCHARS  =  ICHARS  =  0                                               OUTF 270
          CALL INSERTS (IPUNCT(4), 1, LCHARS, LSTATE(1), 1)                     OUTF 280
          CALL INSERTS (A1, 1, LCHARS, LSTATE(1), 3)                            OUTF 290
          CALL INSERTS (IPUNCT(3), 1, LCHARS, LSTATE(1), 2)                     OUTF 300
          GO TO 1060                                                            OUTF 310
C     RETREVIE THE FORMAT STATEMENT FROM THE ARRAY KFOUT.                       OUTF 320
 1010     IN  =  KFOUT (2, JJ)                                                  OUTF 330
          LCHARS  =  ICHARS  =  KFOUT (3, JJ)                                   OUTF 340
          NFOUT  =  NFOUT - 1                                                   OUTF 350
          DO 1030  I  =  1, 3                                                   OUTF 360
            IF (NFOUT .LT. JJ)  GO TO 1030                                      OUTF 370
              DO 1020  JJJ  =  JJ, NFOUT                                        OUTF 380
 1020         KFOUT (I, JJJ)  =  KFOUT (I, JJJ + 1)                             OUTF 390
 1030     KFOUT (I, NFOUT + 1)  =  0                                            OUTF 400
          IPOINT  =  1                                                          OUTF 410
            DO 1040  II  =  IN, 1000, 10                                        OUTF 420
            I2  =  MIN0 (IPOINT + 99, ICHARS)                                   OUTF 430
            IC  =  I2 + 1 - IPOINT                                              OUTF 440
            IF (IC .LE. 0)  GO TO 1050                                          OUTF 450
            DECODE (IC,   20, LFOUT (II))   (LSTATE (I), I=IPOINT, I2)          OUTF 460
 1040       IPOINT  =  IPOINT + 100                                            OUTF 470
C     COMPLETE  THE FORMAT STATEMENT.                                           OUTF 480
 1050     CALL INSERTS (IBLANK, ICHARS + 1, LCHARS, LSTATE(1), I)               OUTF 490
          CALL INSERTS (IPUNCT(4), ICHARS + 1, LCHARS, LSTATE(1), 1)            OUTF 500
          CALL INSERTS (IPUNCT(3), 1, LCHARS, LSTATE(1), 2)                     OUTF 510
 1060     CALL INSERTS (FORMAT, 1, LCHARS, LSTATE(1), 8)                        OUTF 520
          CALL INSERTS (IBLANK, 1, LCHARS, LSTATE(1), 2)                        OUTF 530
          CALL INSERTN (J * 10, 1, LCHARS, LSTATE(1), 4)                        OUTF 540
          II  =  18                                                             OUTF 550
```

```
      SUBROUTINE  OUTFRM                          CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                C   START HERE TO SPACE OUT THE BALANCE OF THE FORMAT STATEMENT.          OUTF 560
                 I070     IF (II .GE. LCHARS)  GO TO I240                               OUTF 570
                          IS  =  II                                                     OUTF 580
                C   SEARCH FOR THE FIRST SPECIAL CHARACTER.                            OUTF 590
60                        DO I090  II  =  IS, LCHARS                                    OUTF 600
                C   IS THE FIRST SPECICAL CHARACTER A /                                OUTF 610
                          IF (LSTATE(II) .EQ. IPUNCT(I))  GO TO I200                    OUTF 620
                C   IS THE FIRST SPECICAL CHARACTER A ,                                OUTF 630
                          IF (LSTATE(II) .EQ. IPUNCT(2))  GO TO I220                    OUTF 640
65              C   IS THE FIRST SPECICAL CHARACTER A * OR A ≠                         OUTF 650
                            DO 1080  JJ  =  5, II, 6                                    OUTF 660
                            IF (LSTATE(II) .EQ. IPUNCT(JJ))  GO TO II70                 OUTF 670
                 I080       CONTINUE                                                    OUTF 680
                          IF (LSTATE(II) .EQ. H)  GO TO 1I00                            OUTF 690
70               I090     CONTINUE                                                     OUTF 700
                        GO TO I240                                                      OUTF 710
                C   NO IT IS AN H.                                                     OUTF 720
                C    IS THIS A HOLERITH FIELD                                         OUTF 730
                 I1O0     IPR  =  II - 2                                              OUTF 740
75                          DO III0  I  =  1, I0                                        OUTF 750
                            IF (INTEGER(I) .EQ. LSTATE(II-I))  GO TO 1I20               OUTF 760
                 III0       CONTINUE                                                    OUTF 770
                C    NO, REPEAT THE  SEARCH                                           OUTF 780
                          II  =  II + I                                                 OUTF 790
80                      GO TO I070                                                      OUTF 800
                C   DETERMINE THE LENGTH OF THE HOLERITH FIELD.                        OUTF 810
                 1I20     N  =  I - I                                                  OUTF 820
                            DO II30  I  =  I, 10                                        OUTF 830
                            IF (INTEGER(I) .EQ. LSTATE(II-2))  GO TO 1I40               OUTF 840
85               II30       CONTINUE                                                    OUTF 850
                        GO TO 1I50                                                      OUTF 860
                 II40     N  =  N + I0 * (I - I)                                       OUTF 870
                          IPR  =  IPR - I                                               OUTF 880
                          IF  (INTEGER(2) .EQ. LSTATE(II - 3))  N  =  N + I00           OUTF 890
90                        IF  (N .GE. 100)  IPR  =  IPR - I                           OUTF 900
                 II50     IF (LSTATE(IPR) .EQ. IBLANK)  GO TO II60                     OUTF 910
                          CALL INSERTS (IBLANK, IPR, LCHARS, LSTATE(1), I)              OUTF 920
                          II  =  II + I                                                 OUTF 930
                 I160     ILAST  =  II + N                                             OUTF 940
95                        IFIRST  =  II + I                                           OUTF 950
                        GO TO II80                                                      OUTF 960
                C   INSERT A BLANK BEFORE AN * OR ≠ AND THEN SKIP TO THE NEXT * OR ≠.  OUTF 970
                 I170     CALL INSERTS (IBLANK, II, LCHARS, LSTATE(I), I)             OUTF 980
                          IFIRST  =  II + 2                                             OUTF 990
I00                       ILAST  =  ISCANL (IPUNCT(JJ), IFIRST, LCHARS, LSTATE(I))    OUTFI000
                 II80     II  =  ILAST + I                                            OUTFI010
                C   ALTER THE HOLERITH FIELDS TO ASSURE PROPER OUTPUT SPACING.        OUTFI020
                            DO II90  I  =  IFIRST, ILAST                                OUTFI030
                 I190       LSTATE (I)  =  LSTATE (I) + 1                             OUTFI040
105                       IF (II .GE. LCHARS)  GO TO I240                             OUTFI050
                          IF (LSTATE(II) .EQ. IPUNCT(I))  GO TO I200                    OUTFI060
                          IF  (LSTATE(II) .EQ. IPUNCT(2))  II  =  II + I                OUTFI070
                          GO TO 1230                                                    OUTFI080
                C    INSERT A BLANK BEFORE THE FIRST AND AFTER THE LAST /.             OUTFI090
I10              I200     CALL INSERTS (IBLANK, II, LCHARS, LSTATE(I), I)             OUTFII00
```

```
   SUBROUTINE  OUTFRM                                CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                      I1  =  I1 + 2                                                        OUTF1110
                1210  IF (LSTATE(I1) .NE. IPUNCT(1))  GO TO 1230                         OUTF1120
                      I1  =  II + 1                                                        OUTF1130
                      GO TO 1210                                                           OUTF1140
115             C    INSERT A BLANK AFTER A COMMA.                                        OUTF1150
                1220  I1  =  I1 + 1                                                       OUTF1160
                C   INSERT A BLANK                                                       OUTF1170
                1230  CALL INSERTS (IBLANK, I1, LCHARS, LSTATE(1), 1)                    OUTF1180
                      II  =  I1 + 1                                                        OUTF1190
120                   GO TO 1070                                                         OUTF1200
                1240  CALL PUNCHIT (17)                                                  OUTF1210
                    LCHARS  =  1                                                         OUTF1220
                    LSTATE (1)  =  C                                                     OUTF1230
                    CALL PUNCHIT (0)                                                     OUTF1240
125             9999  RETURN                                                           OUTF1250
                C                                                                        OUTF1260
                  10  FORMAT  (  *0COULD NOT FIND FORMAT NUMBER * 15,  * IN THE ARRAY *OUTF1270
                    1   *KFOUT.  A DUMMY FORMAT STATEMENT (A1) WAS INSERTED. *  )       OUTF1280
                  20  FORMAT  ( 100A1 )                                                 OUTF1290
130             C                                                                        OUTF1300
                      END                                                                OUTF1310
```

SUBROUTINE OUTPUT CDC 6600 FTN V3.0-P355 OPT=1 06/25/75 12.55.39.

```
      SUBROUTINE  OUTPUT (LIST)                                          OUTP  10
C     THIS ROUTINE WRITES THE WORK FILE RECORD FOR EACH ROUTINE STATEMENT OUTP  20
      COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,     OUTP  30
     1   IPROG, ISNUM, ITYPE, 19999, KFORM (100), KFOUT (3, 100), KSNUM  OUTP  40
     2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),      OUTP  50
     3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,   OUTP  60
     4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING  OUTP  70
     5   (2, 100)                                                       OUTP  80
      COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT      OUTP  90
     1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,     OUTP 100
     2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,     OUTP 110
     3   STAR, X                                                        OUTP 120
      DIMENSION   LIST (1), LOUT (200)                                   OUTP 130
        DO 1000  I  =  1, 100                                            OUTP 140
 1000   LOUT (I)  =  IBLANK                                              OUTP 150
        DO 1010  II  =  1, 200, 10                                       OUTP 160
        I1  =  II * 10 - 9                                               OUTP 170
        I2  =  MIN0 (I1 + 99, ICHARS)                                    OUTP 180
        NC  =  I2 + 1 - I1                                               OUTP 190
        IF (NC .LE. 0)  GO TO 1020                                       OUTP 200
 1010   ENCODE (NC,   10, LOUT (I1))   (LIST (I), I=I1, I2)             OUTP 210
 1020  LWORDS  =   (ICHARS + 9) / 10                                     OUTP 220
      WRITE (LUSTATE)  ITYPE,LWORDS, ICHARS, ISNUM, (LOUT(I), I=1,      OUTP 230
     1   LWORDS), NUMIN, ( (INNUM(I, J), I=1, 2), J=1, NUMIN)           OUTP 240
      NOUTS  =  NOUTS + 1                                                OUTP 250
 9999 RETURN                                                             OUTP 260
C                                                                        OUTP 270
   10  FORMAT  ( 100A1 )                                                OUTP 280
C                                                                        OUTP 290
      END                                                                OUTP 300
```

SUBROUTINE OUTSTR CDC 6600 FTN V3.0-P355 OPT=1 06/25/75 12.55.39.

```
      SUBROUTINE  OUTSTR                                                OUTS  10
C     THIS ROUTINE SETUP THE FINAL DIMENSION AND TYPED STATEMENT RECORDS.OUTS  20
      COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,    OUTS  30
     I   IPROG, ISNUM, ITYPE, 19999, KFORM (100), KFOUT (3, 100), KSNUM OUTS  40
     2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),     OUTS  50
     3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  OUTS  60
     4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING OUTS  70
     5   (2, 100)                                                       OUTS  80
      COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT     OUTS  90
     I   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,     OUTS 100
     2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,     OUTS 110
     3   STAR, X                                                        OUTS 120
      DIMENSION   KTYPE (7)                                             OUTS 130
      INTEGER     STRING, TEST (20)                                     OUTS 140
      DATA    KTYPE / 9HDIMENSION, 8HEXTERNAL, 7HCOMPLEX, 6HDOUBLE,     OUTS 150
     I   7HINTEGER, 7HLOGICAL, 4HREAL  /                                OUTS 160
      NE  =  0                                                          OUTS 170
        DO 1020  J  =  1, 7                                             OUTS 180
C       SKIP THE TYPE IF NONE OCCUR                                     OUTS 190
        IF (NUMBER(J) .EQ. 0)  GO TO 1020                               OUTS 200
C     INSERT THE TYPE NAME.                                             OUTS 210
          N  =  NUMBER (J)                                              OUTS 220
          LCHARS  =  7                                                  OUTS 230
          CALL INSERTS (KTYPE(J), 8, LCHARS, LSTATE(1), 10)             OUTS 240
          LCHARS  =  19                                                 OUTS 250
          IPOINT  =  20                                                 OUTS 260
C     INSERT ONE VARIABLE AT A TIME.                                    OUTS 270
            DO 1000  K  =  I, N                                         OUTS 280
            NE  =  NE + I                                               OUTS 290
            DECODE (20,   10, STRING (1, NE))  TEST                     OUTS 300
            NN  =  NONR (IBLANK, I, 20, TEST(1))                        OUTS 310
            CALL INSERTS (STRING(1, NE), IPOINT, LCHARS, LSTATE(I), NN) OUTS 320
            IPOINT  =  LCHARS + 1                                       OUTS 330
            IF (K .EQ. N)  GO TO 1010                                   OUTS 340
            CALL INSERTS (IPUNCT(2), IPOINT, LCHARS, LSTATE(1), 2)      OUTS 350
            IPOINT  =  LCHARS + I                                       OUTS 360
 1000       CONTINUE                                                    OUTS 370
 1010     CALL PUNCHIT (J + 6)                                         OUTS 380
          NUMBER (J)  =  0                                              OUTS 390
 1020     CONTINUE                                                     OUTS 400
 9999   RETURN                                                         OUTS 410
C                                                                       OUTS 420
   10   FORMAT  ( 100A1 )                                             OUTS 430
C                                                                       OUTS 440
        END                                                             OUTS 450
```

```
      SUBROUTINE  PUNCHIT (ITY)                                          PUNC  10
C     THIS ROUTINE WRITES THE REORGANIZED STATEMENTS ON THE OUTPUT FILE  PUNC  20
C       TAPE4.  THIS FILE IS READY FOR COMPILATION OR PUNCHING.          PUNC  30
        COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,   PUNC  40
     1    IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM PUNC  50
     2    (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),     PUNC  60
     3    LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  PUNC  70
     4    NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING PUNC  80
     5    (2, 100)                                                       PUNC  90
      COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT      PUNC 100
     1    (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,     PUNC 110
     2    MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,     PUNC 120
     3    STAR, X                                                        PUNC 130
      DIMENSION   LINEOUT (72)                                          PUNC 140
      INTEGER     C, H, STAR, X                                         PUNC 150
      IF (ITY .EQ. 16)  CALL FIXDATA                                    PUNC 160
      IPOINT  =  72                                                     PUNC 170
      NNN  =  7                                                         PUNC 180
        DO 1000  I  =  1, 72                                            PUNC 190
 1000   LINEOUT (I)  =  LSTATE (I)                                      PUNC 200
      NCARDS  =  NCARDS + 1                                             PUNC 210
      IF (NCARDS .LT. 1000)  GO TO 1010                                 PUNC 220
      ICOUNT (1, 2)  =  ICOUNT (1, 2) + 99                              PUNC 230
      NAME (4)  =  IPUNCT (5)                                           PUNC 240
      NCARDS  =  1                                                      PUNC 250
 1010 CONTINUE                                                          PUNC 260
C     ONE OR THE FIRST CARD OUTPUT.                                     PUNC 270
      IF (LCHARS .LE. 72)  GO TO 1080                                   PUNC 280
      IF (ITY .EQ. 16)  GO TO 1020                                      PUNC 290
      NNN  =  10                                                        PUNC 300
      IF (LSTATE(73) .EQ. IBLANK .OR. LINEOUT(72) .EQ. IBLANK)  GO TO   PUNC 310
     1  1080                                                           PUNC 320
C     FIND THE LAST BLANK FOR THE BREAK LOCATION.                       PUNC 330
      N  =  ISCANR (IBLANK, 62, 72, LINEOUT(1))                         PUNC 340
      IF (N .GE. 62 .AND. N .LE. 72)  GO TO 1030                        PUNC 350
      IF (ITY .NE. 17)  GO TO 1080                                      PUNC 360
C     WITH A FORMAT OR A DATA STATEMENT BREAK ONLY AFTER A COMMA, /, OR )PUNC 370
 1020 N2  =  ISCANR (IPUNCT(2), 61, 72, LINEOUT(1)) + 1                 PUNC 380
      N1  =  ISCANR (IPUNCT(1), 61, 72, LINEOUT(1)) + 1                 PUNC 390
      N4  =  ISCANR (IPUNCT(4), 61, 72, LINEOUT(1)) + 1                 PUNC 400
      N  =  MAX0 (N1, N2, N4)                                           PUNC 410
      IF (N .EQ. 73)  GO TO 1070                                        PUNC 420
      NNN  =  7                                                         PUNC 430
      IF (N .LT. 62)  GO TO 1050                                        PUNC 440
      NNN  =  10                                                        PUNC 450
 1030   DO 1040  I  =  N, 72                                           PUNC 460
 1040   LINEOUT (I)  =  IBLANK                                         PUNC 470
      IPOINT  =  N                                                      PUNC 480
      GO TO 1080                                                        PUNC 490
 1050 IF (ITY .EQ. 16)  GO TO 1080                                      PUNC 500
      IF (LINEOUT(72) .EQ. IPUNCT(5))  GO TO 1070                       PUNC 510
C     HERE TO INSERT AN * INTO A FORMAT STATEMENT.                      PUNC 520
      N5  =  ISCANR (IPUNCT(5), 16, 71, LINEOUT(1))                     PUNC 530
      N11  =  ISCANR (IPUNCT(11), 16, 71, LINEOUT(1))                   PUNC 540
      J  =  5                                                           PUNC 550
```

```
   SUBROUTINE  PUNCHIT                                  CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                    IF (N11 .LT. N5)  GO TO 1060                                                  PUNC 560
                    N5  =  N11                                                                    PUNC 570
                    J  =  11                                                                      PUNC 580
               1060 CONTINUE                                                                      PUNC 590
 60                 NH  =  ISCANR (H, 16, 71, LINEOUT(1))                                         PUNC 600
                    IF (NH .GT. N5 .OR. N5 .LT. 16)  GO TO 1080                                   PUNC 610
                    LINEOUT (72)  =  IPUNCT (J)                                                   PUNC 620
                    IPOINT  =  IPOINT - 1                                                         PUNC 630
                    LSTATE (IPOINT)  =  IPUNCT (J)                                                PUNC 640
 65                 IPOINT  =  IPOINT - 1                                                         PUNC 650
               1070 NNN  =  10                                                                    PUNC 660
               1080 WRITE (LUOUT,  10)  LINEOUT, NAME, NCARDS                                     PUNC 670
                    IF (LCHARS .LE. 72)  GO TO 1220                                               PUNC 680
              C     MULTIPLE CARD OUTPUT.                                                         PUNC 690
 70                 IC  =  1                                                                      PUNC 700
              C     INDENT THE REMAINING DATA STRING.                                             PUNC 710
              C       NN      THE STARTING LOCATION FOR THE CONTINUATION CARDS.                   PUNC 720
                    NN  =  10 + 2 * NPUSH                                                         PUNC 730
              C     FORMAT OR DATA STATEMENT, CAN NOT BE INDENTED.                                PUNC 740
 75                 IF  (ITY .EQ. 16 .OR. ITY .EQ. 17)  NN  =  NNN                                PUNC 750
               1090   DO 1100  I  =  1, 72                                                       PUNC 760
               1100   LINEOUT (I)  =  IBLANK                                                     PUNC 770
               1110 IF (LSTATE(IPOINT+1) .NE. IBLANK)  GO TO 1120                                 PUNC 780
                    IPOINT  =  IPOINT + 1                                                         PUNC 790
 80                 GO TO 1110                                                                    PUNC 800
               1120   DO 1130  I  =  NN, 72                                                      PUNC 810
                      IPOINT  =  IPOINT + 1                                                       PUNC 820
               1130   LINEOUT (I)  =  LSTATE (IPOINT)                                           PUNC 830
                    IC  =  MINO (IC + 1, 10)                                                      PUNC 840
 85                 LINEOUT (6)  =  INTEGER (IC)                                                  PUNC 850
                    NCARDS  =  NCARDS + 1                                                         PUNC 860
                    IF (NCARDS .LT. 1000)  GO TO 1140                                             PUNC 870
                    ICOUNT (1, 2)  =  ICOUNT (1, 2) + 99                                          PUNC 880
                    NAME (4)  =  IPUNCT (5)                                                       PUNC 890
 90                 NCARDS  =  1                                                                  PUNC 900
               1140 CONTINUE                                                                      PUNC 910
                    IF (LCHARS .LE. IPOINT)  GO TO 1210                                           PUNC 920
                    IF (ITY .EQ. 16)  GO TO 1150                                                  PUNC 930
                    IF (LSTATE(IPOINT+1) .EQ. IBLANK .OR. LINEOUT(72) .EQ. IBLANK)                PUNC 940
 95                1   GO TO 1210                                                               PUNC 950
              C     FIND THE LAST BLANK FOR THE BREAK LOCATION.                                   PUNC 960
                    N  =  ISCANR (IBLANK, 62, 72, LINEOUT(1))                                     PUNC 970
                    IF (N .GE. 62 .AND. N .LE. 72)  GO TO 1160                                    PUNC 980
                    IF (ITY .NE. 17)  GO TO 1210                                                  PUNC 990
100           C     WITH A FORMAT OR A DATA STATEMENT BREAK ONLY AFTER A COMMA, /, OR )PUNC1000
               1150 N2  =  ISCANR (IPUNCT(2), 61, 72, LINEOUT(1)) + 1                             PUNC1010
                    N1  =  ISCANR (IPUNCT(1), 61, 72, LINEOUT(1)) + 1                             PUNC1020
                    N4  =  ISCANR (IPUNCT(4), 61, 72, LINEOUT(1)) + 1                             PUNC1030
                    N  =  MAX0 (N1, N2, N4)                                                       PUNC1040
105                 IF (N .EQ. 73)  GO TO 1200                                                    PUNC1050
                    NN  =  7                                                                      PUNC1060
                    IF (N .LT. 62)  GO TO 1180                                                    PUNC1070
                    NN  =  10                                                                     PUNC1080
               1160   DO 1170  I  =  N, 72                                                       PUNC1090
110            1170   LINEOUT (I)  =  IBLANK                                                     PUNC1100
```

```
   SUBROUTINE  PUNCHIT                                CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                      IPOINT  =  N + IPOINT - 72                                          PUNC1110
                      GO TO 1210                                                          PUNC1120
                1180  IF (ITY .EQ. 16)  GO TO 1210                                        PUNC1130
                      IF (LINEOUT(72) .EQ. IPUNCT(5))  GO TO 1200                         PUNC1140
115             C     HERE TO INSERT AN * INTO A FORMAT STATEMENT.                        PUNC1150
                      N5  =  ISCANR (IPUNCT(5), NN, 71, LINEOUT(1))                       PUNC1160
                      N11  =  ISCANR (IPUNCT(11), NN, 71, LINEOUT(1))                     PUNC1170
                      J  =  5                                                             PUNC1180
                      IF (N11 .LT. N5)  GO TO 1190                                        PUNC1190
120                   J  =  11                                                            PUNC1200
                      N5  =  N11                                                          PUNC1210
                1190  CONTINUE                                                            PUNC1220
                      NH  =  ISCANR (H, NN, 71, LINEOUT(1))                               PUNC1230
                      IF (NH .GT. N5 .OR. N5 .LT. NN)  GO TO 1210                         PUNC1240
125                   LINEOUT (72)  =  IPUNCT (J)                                         PUNC1250
                      IPOINT  =  IPOINT - 1                                               PUNC1260
                      LSTATE (IPOINT)  =  IPUNCT (J)                                      PUNC1270
                      IPOINT  =  IPOINT - 1                                               PUNC1280
                1200  NN  =  10                                                           PUNC1290
130             1210  WRITE (LUOUT,  10)  LINEOUT, NAME, NCARDS                          PUNC1300
                      IF (IPOINT .LT. LCHARS)  GO TO 1090                                 PUNC1310
                C                                                                         PUNC1320
                1220    DO 1230  I  =  1, LCHARS                                          PUNC1330
                1230    LSTATE (I)  =  IBLANK                                             PUNC1340
135                   LCHARS  =  0                                                        PUNC1350
                9999  RETURN                                                              PUNC1360
                C                                                                         PUNC1370
                  10  FORMAT  ( 76A1, I3,  *0*  )                                        PUNC1380
                C                                                                         PUNC1390
140                   END                                                                 PUNC1400
```

```
      SUBROUTINE  READS                                                  READ  10
C     THIS ROUTINE READS THE INPUT FILE AND GENERATES THE WORK FILE  AND READ  20
C       STRINGS FOR LATER PROCESSING.                                    READ  30
        COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,   READ  40
     1    IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM READ  50
     2    (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),     READ  60
     3    LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  READ  70
     4    NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING READ  80
     5    (2, 100)                                                       READ  90
        COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT    READ 100
     1    (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,     READ 110
     2    MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,     READ 120
     3    STAR, X                                                        READ 130
        DIMENSION   NAMES (4,2)                                         READ 140
        INTEGER     C, END, GOTO, PROGRAM, STAR, STRING, TRANSF         READ 150
        LOGICAL     CHECK, KLIST, KO                                    READ 160
        DATA    NAMES / 1HN, 1HA, 1HM, 1HE, 1HD, 1HA, 1HT, 1HA  /      READ 170
        DATA    GOTO / 5HGO TO  /                                       READ 180
        ICOUNT (I, I)  =  I                                             READ 190
        IF (NCARD .NE. 0)  GO TO 1020                                   READ 200
        ICOUNT (1, 1)  =  0                                             READ 210
 1000   READ (LUIN,   10)  LCARD                                        READ 220
        ICOUNT (I, 1)  =  ICOUNT (I, 1) + I                             READ 230
        IF (EOF(LUIN))   1860, 1010                                     READ 240
 1010   NCARD  =  I                                                     READ 250
C     CHECK FOR A COMMENT CARD, IF SO OUTPUT.                           READ 260
 1020   IF (IEOF .EQ. I)  GO TO 9999                                    READ 270
        IF (LCARD(I) .EQ. C .OR. LCARD(1) .EQ. STAR)  GO TO 1030        READ 280
C     CHECK FOR AN ALL BLANK CARD, IF SO OUTPUT C CARD.                 READ 290
        IF (NONL(IBLANK,I,72,LCARD(I)) .LE. 72)  GO TO 1050             READ 300
        ICHARS  =  I                                                    READ 310
        GO TO 1040                                                      READ 320
 1030   ICHARS  =  NONR (IBLANK, 2, 72, LCARD(I))                       READ 330
 1040   ITYPE  =  0                                                     READ 340
        LCARD (I)  =  C                                                 READ 350
        CALL OUTPUT (LCARD(1))                                          READ 360
        NCARD  =  0                                                     READ 370
        GO TO 1000                                                      READ 380
C     CHECK FOR A STATEMENT NUMBER IN THE FIRST 5 COLUMNS.              READ 390
 1050   N  =  NONL (IBLANK, I, 5, LCARD(I))                             READ 400
        IF (N .GT. 5)  GO TO 1060                                       READ 410
C     YES.  NOW DETERMINE ITS VALUE                                     READ 420
        ISTOP  =  5                                                     READ 430
        NVALUE  =  NUMBS (N, ISTOP, LCARD(I))                           READ 440
        IF (NVALUE .GT. 0)  GO TO 1060                                  READ 450
        PRINT   20, LCARD                                               READ 460
        GO TO 1030                                                      READ 470
C     TRANSFER THIS RECORD TO LSTATE.                                   READ 480
 1060   ITRANS  =  TRANSF (7, 72)                                       READ 490
 1070   IF (ITRANS .GT. 0)  GO TO 1100                                  READ 500
C     READ THE NEXT INPUT RECORD.                                       READ 510
        READ (LUIN,   10)  LCARD                                        READ 520
        ICOUNT (I, 1)  =  ICOUNT (I, I) + I                             READ 530
        IF (EOF(LUIN))   1090, 1080                                     READ 540
 1080   NCARD  =  I + NCARD                                             READ 550
```

```
      C     CHECK ZF A CONTINUATION RECORD.                                    READ 560
            IF (LCARD(6) .EQ. INTEGER(1) .OR. LCARD(6) .EQ. IBLANK .OR.        READ 570
           1   LCARD(1) .EQ. C .OR. LCARD(1).EQ.STAR)  GO TO 1100              READ 580
      C     YES, THEN SET UP FOR THE TRANSFER TO LSTATE.                       READ 590
   60       N  =  NONL (IBLANK, 1, 5, LCARD(1))                                READ 600
            IF (N .GT. 5)  GO TO 1060                                          READ 610
            GO TO 1100                                                         READ 620
      C     THE ENTIRE ARRAY HAS BEEN CONSTRUCTED, NOW IDENTIFY THE TYPE AND   READ 630
      C       INSERT THE PROPER SPACING.                                       READ 640
   65  1090  IEOF  =  1                                                        READ 650
            ICOUNT (1, 1)  =  ICOUNT (1, 1) - 1                                READ 660
       1100 IF (IPROG .NE. 0)  GO TO 1200                                      READ 670
            CALL BLANKS                                                        READ 680
      C     DO PROGRAM STATEMENTS ITYPE = 1, 2, 3, 4.                          READ 690
   70       IPOINT  =  1                                                       READ 700
            J  =  IDENT (1)                                                    READ 710
            IF (J .NE. 45)  GO TO 1120                                         READ 720
      C     NO ROUTINE TYPE CARD FOUND, THIS IS AN ERROR.                      READ 730
            PRINT  30                                                          READ 740
   75       IPROG  =  100                                                      READ 750
              DO 1110  I  =  1, 4                                              READ 760
       1110   NAME (I)  =  NAMES (I, 1)                                        READ 770
            GO TO 1200                                                         READ 780
      C     A ROUTINE TYPE MATCH WAS FOUND.                                    READ 790
   80  1120 ITYPE  =  J                                                       READ 800
            IPROG  =  J                                                        READ 810
            IF (J .NE. 4)  GO TO 1140                                          READ 820
      C     HERE FOR BLOCK DATA.                                               READ 830
            CALL INSERT (IBLANK, IPOINT - 4, LCHARS, LSTATE(1), 1)             READ 840
   85       IPOINT  =  IPOINT + 1                                              READ 850
      C     IS THE BLOCK DATA NAMED.                                           READ 860
            IF (IPOINT .LE. ICHARS)  GO TO 1140                                READ 870
      C     NO ,USE *DATA*.                                                    READ 880
              DO 1130  I  =  1, 4                                              READ 890
   90  1130   NAME (I)  =  NAMES (1, 2)                                        READ 900
            GO TO 1690                                                         READ 910
      C     FINALLY  SETUP THE NAME.                                           READ 920
       1140 CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), 2)                 READ 930
            IPOINT  =  IPOINT + 2                                              READ 940
   95 C     OBTAIN THE ROUTINE NAME FOR LATER USE IN THE OUTPUT CARD COL 73-76.READ 950
            IP  =  IPOINT                                                      READ 960
              DO 1150  I  =  1, 4                                              READ 970
              IF (LSTATE(IP) .EQ. IPUNCT(3))  GO TO 1160                       READ 980
              NAME (I)  =  LSTATE (IP)                                         READ 990
  100  1150   IP  =  IP + 1                                                    READ1000
       1160 CONTINUE                                                         READ1010
            IF (IPROG .NE. 1)  GO TO 1680                                      READ1020
              DO 1170  I  =  1, 4                                              READ1030
       1170   ICOUNT (2, I)  =  0                                             READ1040
  105          DO 1180  1  =  1, 7                                            READ1050
       1180   PROGRAM (I)  =  IBLANK                                          READ1060
            IP  =  IPOINT                                                      READ1070
              DO 1190  I  =  1, 7                                              READ1080
              IF (LSTATE(IP) .EQ. IPUNCT(3))  GO TO 1680                       READ1090
  110         PROGRAM (I)  =  LSTATE (IP)                                      READ1100
```

```
 1190    IP  =  IP + 1                                                  READ1110
         GO TO 1680                                                     READ1120
C   START PROCESSING THE ROUTINE STATEMENTS.                           READ1130
 1200  IPOINT  =  1                                                    READ1140
115      J  =  IDENT (2)                                               READ1150
         IERROR  =  J                                                  READ1160
         ITYPE  =  J                                                   READ1170
         IF (J .LE. 4 .OR. J .GT. 45)  GO TO 1800                      READ1180
         CALL BLANKS                                                   READ1190
120      GO TO  ( 1240, 1220, 1260, 1260, 1220, 1220, 1220, 1220, 1220,  READ1200
        1   1250, 1270, 1210, 1290, 1310, 1320, 1230, 1380, 1220, 1460,  READ1210
        2   1700, 1470, 1230, 1230, 1470, 1230, 1490, 1500, 1470, 1470,  READ1220
        3   1220, 1220, 1220, 1570, 1220, 1220, 1580, 1220, 1220, 1620,  READ1230
        4   1790, 1640)  J - 4                                          READ1240
125   C    MAKE A SPECIAL CHECK FOR DATA STATEMENTS. J = 16.            READ1250
      C      DATA (TEXT(I),I=1,9) / LIST   IS OK, MUST CHECK FOR THE RELATIVE READ1260
      C      POSITIONS OF THE MATCHING ( ), I4, AND THE =, I8.         READ1270
 1210  IF (LSTATE(IPOINT) .NE. IPUNCT(3))  GO TO 1230                  READ1280
         I4  =  MATCH (IPOINT, ICHARS, LSTATE(1))                      READ1290
130      I8  =  ISCANL (IPUNCT(8), IPOINT + 1, ICHARS, LSTATE(1))      READ1300
         IF (I8 .LT. I4)  GO TO 1280                                   READ1310
         GO TO 1630                                                    READ1320
      C    CHECK FOR ( OR = FOLLOWING THE TYPE WORD JUST IDENTIFIED.   READ1330
 1220  IF (LSTATE(IPOINT) .EQ. IPUNCT(3))  GO TO 1630                  READ1340
135   1230  IF (LSTATE(IPOINT) .EQ. IPUNCT(8))  GO TO 1630             READ1350
      C    NOW WORK THE STATEMENTS.                                    READ1360
         GO TO  ( 1240, 1620, 1260, 1260, 1260, 1260, 1260, 1260, 1260,  READ1370
        1   1250, 1270, 1280, 1290, 1310, 1320, 1360, 1380, 1450, 1460,  READ1380
        2   1700, 1470, 1480, 1480, 1470, 1480, 1490, 1500, 1470, 1470,  READ1390
140     3   1520, 1520, 1530, 1570, 1580, 1580, 1580, 1590, 1580, 1620,  READ1400
        4   1790, 1640)  J - 4                                          READ1410
      C                                                                 READ1420
 1240  CALL INSERT (IBLANK, IPOINT - 1, LCHARS, LSTATE(1), 2)           READ1430
         IPOINT  =  1 + ISCANL (IPUNCT(1), IPOINT + 3, ICHARS, LSTATE(1))  READ1440
145      GO TO 1620                                                    READ1450
      C    SET PRECISION TO DOUBLE.                                    READ1460
 1250  J  =  10                                                        READ1470
      C    STORE THE TYPE STATEMENTS IN THE ARRAY STRING.              READ1480
 1260  CALL STORE (J - 6)                                              READ1490
150      GO TO 1730                                                    READ1500
      C    EQUIVALANCE STATEMENTS INSERT A BLANK (15).                 READ1510
 1270  CALL INSERT (IBLANK, IPOINT - 1, LCHARS, LSTATE(1), 1)          READ1520
         GO TO 1680                                                    READ1530
      C    DO DATA   STATEMENTS J = 16.                                READ1540
155   1280  CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), 4)          READ1550
         IPOINT  =  IPOINT + 4                                         READ1560
         GO TO 1680                                                    READ1570
      C    DO FORMAT STATEMENTS  J  =  17.                             READ1580
 1290  ICHARS  =  ICHARS - IPOINT                                      READ1590
160      IF ( .NOT. KO(NVALUE))  GO TO 1730                            READ1600
         IN  =  KFOUT (2, NFOUT)                                       READ1610
           DO 1300  II  =  IN, 1000, 10                                READ1620
           I2  =  MIN0 (IPOINT + 99, LCHARS - 1)                       READ1630
           IC  =  I2 + 1 - IPOINT                                      READ1640
165        IF (IC .LE. 0)  GO TO 1730                                  READ1650
```

```
SUBROUTINE  READS                                   COC 6600 FTN V3.0-P355 OPT=1   06/25/75  12.55.39.

                ENCODE (IC,   10, LFOUT (11))   (LSTATE (I), I=IPOINT, 12)      READ1660
          I300  IPOINT  =  1POINT + 100                                         READ1670
                GO TO I730                                                      READ1680
          C  J  =  I8  STATEMENT TYPE CO 999  I  =  9, 99                       READ1690
170       I310  N  =  NUMBS (IPOINT, LCHARS, LSTATE(1))                        READ1700
                IF (N .LE. 0)  GO TO 1630                                       READ1710
                IF ( .NOT. KLIST(IPOINT,N))  GO TO 1640                         READ1720
                CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(I), 2)              READ1730
                GO TO I640                                                      READ1740
175       C    J  =  19  STATEMENT TYPE GO TO  (9999,9999) V                    READ1750
          1320  CALL INSERT (IBLANK, IPOINT - 1, LCHARS, LSTATE(I), 2)          READ1760
                CALL INSERT (IBLANK, IPOINT - 3, LCHARS, LSTATE(1), 1)          READ1770
                IPOINT  =  IPOINT + 3                                           READ1780
                N  =  NUMBS (IPOINT, LCHARS, LSTATE(1))                         READ1790
180       C    THERE MUST BE STATEMENT NUMBER IN THE FIRST POSITION.            READ1800
                IF (N .LE. 0)  GO TO 1630                                       READ1810
                GO TO 1340                                                      READ1820
          1330  N  =  NUMBS (IPOINT, LCHARS, LSTATE(1))                        READ1830
                IF (N .LE. 0)  GO TO I350                                       READ1840
185       1340  IF ( .NOT. KLIST(IPOINT,N))  GO TO 1620                       READ1850
                IPOINT  =  IPOINT + 1                                           READ1860
                IF (LSTATE(IPOINT-1) .EQ. IPUNCT(4))  GO TO 1350                READ1870
                GO TO I330                                                      READ1880
          1350  IF  (LSTATE(IPOINT) .EQ. IPUNCT(2))  IPOINT  =  IPOINT + 1     READ1890
190             GO TO 1620                                                      READ1900
          C    J  =  20  STATEMENT TYPE GO TO  9999                            READ1910
          I360  CALL INSERT (IBLANK, IPOINT - 2, LCHARS, LSTATE(1), 1)          READ1920
                IPOINT  =  IPOINT + 1                                           READ1930
                N  =  NUMBS (IPOINT, LCHARS, LSTATE(I))                         READ1940
195             IF (N .GT. 0)  GO TO 1370                                       READ1950
          C    DO THE ASSIGNED GO TO X (9,99,999,9999)                          READ1960
                CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), 2)              READ1970
                IPOINT  =  IPOINT + 2                                           READ1980
                IPOINT  =  ISCANL (IPUNCT(3), IPOINT, LCHARS, LSTATE(I))        READ1990
200             CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), 1)              READ2000
                IPOINT  =  IPOINT + 2                                           READ2010
                GO TO 1330                                                      READ2020
          1370  IF ( .NOT. KLIST(IPOINT,N))  GO TO I680                       READ2030
                ISTOP  =  ICHARS  =  INNUM (1, 1) - 1                           READ2040
205             GO TO 1690                                                      READ2050
          C    J  =  21  STATEMENT TYPE IF (V) ***                             READ2060
          1380  CALL INSERT (IBLANK, IPOINT - 1, LCHARS, LSTATE(1), 1)         READ2070
                IPOINT  =  1 + MATCH (IPOINT, LCHARS, LSTATE(1))                READ2080
                CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), 2)              READ2090
210             IPOINT  =  IPOINT + 2                                           READ2100
                N  =  NUMBS (IPOINT, LCHARS, LSTATE(1))                         READ2110
                IF (N .LE. 0)  GO TO 1410                                       READ2120
                GO TO 1400                                                      READ2130
          1390  IPOINT  =  IPOINT + 1                                          READ2140
215             N  =  NUMBS (IPOINT, LCHARS, LSTATE(1))                         READ2150
                IF (N .LE. 0)  GO TO 1690                                       READ2160
          1400  IF (KLIST(IPOINT,N))  GO TO I390                               READ2170
                GO TO I680                                                      READ2180
          1410  JJ  =  IDENT (2)                                                READ2190
220             IF (JJ .LE. I8 .OR. JJ .GT. 45)  GO TO 1800                     READ2200
```

```
      GO TO  ( 1320, I440, I380, I430, I460, I700, I470, I440, 1440,    READ2210
     1   I470, 1440, I430, I500, I470, I470, I430, I430, I430, I420,    READ2220
     2   I430, I430, 1580, I430, I430, 1620, 1790, 1640)  JJ - 18        READ2230
 1420  IF (LSTATE(IPOINT) .EQ. IPUNCT(5))  GO TO I780                   READ2240
C     CHECK FOR A, (, OR A, =, FOLLOWING THE IDENTIFIES NAME.           READ2250
 I430  IF (LSTATE(IPOINT) .EQ. IPUNCT(3))  GO TO 1640                  READ2260
 I440  IF (LSTATE(IPOINT) .EQ. IPUNCT(8))  GO TO 1640                  READ2270
C                                                                      READ2280
      GO TO  ( 1320, 1360, I380, I450, I460, I700, 1470, 1480, I480,   READ2290
     I   1470, I480, 1490, I500, I470, 1470, I520, I520, I550, 1570,   READ2300
     2   1580, 1580, 1580, I590, 1580, I620, I790, 1640)  JJ - 18       READ2310
C                                                                      READ2320
 1450  CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(I), 1)               READ2330
       IPOINT  =  IPOINT + 2                                           READ2340
       GO TO I680                                                      READ2350
C     J  =  23  STATEMENT TYPE ASSIGN 9999 TO V                        READ2360
 I460  N  =  NUMBS (IPOINT, LCHARS, LSTATE(I))                         READ2370
       IF (N .LE. 0)  GO TO I630                                       READ2380
       IF ( .NOT. KLIST(IPOINT,N))  GO TO I680                         READ2390
       CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), 1)              READ2400
       IPOINT  =  IPOINT + I                                           READ2410
       IF ( .NOT. CHECK(2HTO,2,IPOINT,ICHARS,LSTATE(I),IPOINT))  GO TO READ2420
     I   I680                                                          READ2430
       CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(I), 2)              READ2440
       GO TO I690                                                      READ2450
C     J  =  25  STATEMENT TYPE READ (XX,YY) LIST                       READ2460
C     J  =  28  STATEMENT TYPE WRITE (XX,YY) LIST                      READ2470
C     J  =  32  STATEMENT TYPE DECODE (XX,YY,V) LIST                   READ2480
C     J  =  33  STATEMENT TYPE ENCODE (XX,YY,V) LIST                   READ2490
 1470  CALL INSERT (IBLANK, IPOINT - I, LCHARS, LSTATE(I), 1)           READ2500
       IPOINT  =  IPOINT + I                                           READ2510
       I  =  I + MATCH (IPOINT, LCHARS, LSTATE(I))                     READ2520
       CALL INSERT (IBLANK, I, LCHARS, LSTATE(1), 2)                   READ2530
       IPOINT  =  ISCANL (IPUNCT(2), IPOINT, ICHARS, LSTATE(I)) + 1    READ2540
C     J  =  26  STATEMENT TYPE READ XX, LIST                           READ2550
C     J  =  27  STATEMENT TYPE PRINT XX, LIST                          READ2560
C     J  =  29  STATEMENT TYPE PUNCH XX, LIST                          READ2570
 I480  N  =  NUMBS (IPOINT, LCHARS, LSTATE(I))                         READ2580
       IF (N .LE. 0)  GO TO I680                                       READ2590
       IF ( .NOT. KLIST(IPOINT,N))  GO TO 1680                         READ2600
       CALL KF (N)                                                     READ2610
       GO TO 1680                                                      READ2620
C     J  =  30  STATEMENT TYPE BUFFER IN  (XX,YY,V) LIST               READ2630
 1490  CALL INSERT (IBLANK, IPOINT - 3, LCHARS, LSTATE(1), I)           READ2640
       GO TO 1510                                                      READ2650
C     J  =  31  STATEMENT TYPE BUFFER OUT (XX,YY,V) LIST               READ2660
 1500  CALL INSERT (IBLANK, IPOINT - 4, LCHARS, LSTATE(I), 1)           READ2670
 1510  CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), 1)              READ2680
       IPOINT  =  IPOINT + 1                                           READ2690
       IPOINT  =  I + MATCH (IPOINT, LCHARS, LSTATE(1))                READ2700
       GO TO 1620                                                      READ2710
C     J  =  34  STATEMENT TYPE STOP                                    READ2720
C     J  =  35  STATEMENT TYPE ENTRY ROUTINE                           READ2730
 1520  CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), I)              READ2740
       GO TO 1690                                                      READ2750
```

```
      C     J  =  36  CHANGE A (RETURN) TO A ( GO TO 9999).                  READ2760
      C       UNLESS THIS IS A NUMBERED RETURN.                              READ2770
       1530   CONTINUE                                                       READ2780
              IF (ICHARS .GT. 6)  GO TO 1610                                 READ2790
280           IF (IDOLLAR .GT. 0)  GO TO 1780                                READ2800
              IPOINT  =  1                                                   READ2810
       1540   LCHARS  =  IPOINT - 1                                          READ2820
      C     CHECK IF THE NEXT RECORD IS AN END STATEMENT.                    READ2830
              IP1  =  7                                                      READ2840
285           IP2  =  72                                                     READ2850
              IF ( .NOT. CHECK(END,3,IP1,IP2,LCARD(1),IP3))  GO TO 1560      READ2860
              IF (NONL(IBLANK,IP3,IP2,LCARD(1)) .GT. IP2)  GO TO 1770        READ2870
              GO TO 1560                                                     READ2880
      C     ENTER HERE FOR AN IF STATEMENT                                   READ2890
290    1550   IF (ICHARS .GE. IPOINT)  GO TO 1620                            READ2900
              IF (IDOLLAR .GT. 0)  GO TO 1620                                READ2910
              IPOINT  =  IPOINT - 6                                          READ2920
              GO TO 1540                                                     READ2930
       1560   CALL INSERTN (9999, IPOINT, LCHARS, LSTATE(1), 4)              READ2940
295           CALL INSERT (GOTO, IPOINT, LCHARS, LSTATE(1), 5)               READ2950
              I9999  =  1                                                    READ2960
              ICHARS  =  LCHARS                                              READ2970
              GO TO 1690                                                     READ2980
      C     J = 37 USE (LFN)                                                 READ2990
300    1570   IPOINT  =  1                                                   READ3000
              GO TO 1680                                                     READ3010
      C     J  =  38, 39, 40, 42  ENDFILE, REWIND, BACKSPACE, OR PAUSE.      READ3020
       1580   CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), 1)             READ3030
              GO TO 1690                                                     READ3040
305   C     J  =  41  SURPRESS THE WORD TYPE.                                READ3050
       1590     DO 1600  I  =  1, 4                                          READ3060
       1600     CALL SHIFTL (IBLANK, 1, LCHARS, LSTATE(1))                   READ3070
              GO TO 1200                                                     READ3080
      C     CHANGE A NUMBERED RETURN TO J = 44.                              READ3090
310    1610   J  =  44                                                       READ3100
              IERROR  =  J                                                   READ3110
              ITYPE  =  J                                                    READ3120
      C     J  =  43  NAMELIST.                                              READ3130
       1620   CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), 2)             READ3140
315           IPOINT  =  IPOINT + 2                                          READ3150
              GO TO 1680                                                     READ3160
      C     J  =  45  REPLACEMENT STATEMENT TYPE X  =  V                     READ3170
       1630   J  =  45                                                       READ3180
              ITYPE  =  J                                                    READ3190
320    1640   IP  =  1                                                       READ3200
      C     CHECK FOR THE EQUAL, =, SIGN.                                    READ3210
              IPOINT  =  ISCANL (IPUNCT(8), IP, ICHARS, LSTATE(1))           READ3220
              IF (IPOINT .LT. ICHARS)  GO TO 1660                            READ3230
              PRINT  40, (LSTATE(I), I=1, LCHARS)                            READ3240
325           GO TO 1670                                                     READ3250
       1650   IPOINT  =  ISCANL (IPUNCT(8), IP, ICHARS, LSTATE(1))           READ3260
              IF (IPOINT .GE. ICHARS)  GO TO 1670                            READ3270
       1660   CALL INSERT (IBLANK, IPOINT + 1, LCHARS, LSTATE(1), 2)         READ3280
              CALL INSERT (IBLANK, IPOINT, LCHARS, LSTATE(1), 2)             READ3290
330           IP  =  IPOINT + 5                                              READ3300
```

```
 SUBROUTINE  READS                                CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                GO TO 1650                                                          READ3310
          1670  IPOINT  =  2                                                       READ3320
              C                                                                     READ3330
          1680  CALL SPACOUT                                                      READ3340
335           C                                                                     READ3350
          1690  IF (NVALUE .LE. 0)  GO TO 1720                                    READ3360
          1700  NSTATN  =  NSTATN + 1                                             READ3370
                IF (NSTATN .GT. MNSTATE)  GO TO 1810                              READ3380
                KSNUM (1, NSTATN)  =  NVALUE                                      READ3390
340             IF (J .EQ. 36)  GO TO 1710                                        READ3400
                NSNUMC  =  NSNUMC + 10                                            READ3410
                ISNUM  =  NSNUMC                                                  READ3420
                KSNUM (2, NSTATN)  =  NSNUMC                                      READ3430
                GO TO 1720                                                        READ3440
345       1710  ISNUM  =  0                                                       READ3450
                KSNUM (2, NSTATN)  =  9999                                        READ3500
                I9999  =  1                                                       READ3510
          1720  CALL OUTPUT (LSTATE(1))                                          READ3520
          1730  CALL RESETX                                                       READ3530
350             NCARD  =  1                                                       READ3540
                IF (IDOLLAR .LE. 0)  GO TO 1750                                   READ3550
              C   HERE FOR A DOLLAR SIGN (MULTIPLE STATEMNTS) SHIFT LEFT AND GO AGAINREAD3560
                LCHARS  =  LCHARS - IDOLLAR                                       READ3570
                ICHARS  =  LCHARS                                                 READ3580
355               DO 1740  I  =  1, LCHARS                                        READ3590
                  LSTATE (I)  =  LSTATE (IDOLLAR + I)                             READ3600
          1740    LSTATE (IDOLLAR + I)  =  0                                      READ3610
                GO TO 1200                                                        READ3620
              C   CLEAR THE ARRAY AND RETURN TO START THE NEXT RECORD.            READ3630
360       1750    DO 1760  I  =  1, LCHARS                                        READ3640
          1760    LSTATE (I)  =  IBLANK                                           READ3650
                LCHARS  =  0                                                      READ3660
                IF (ITRANS .EQ. 0)  GO TO 1020                                    READ3670
                ITRANS  =  TRANSF (ITRANS, 72)                                    READ3680
365             NCARD  =  1                                                       READ3690
                GO TO 1070                                                        READ3700
              C   END PROCESSING FOLLOWING A RETURN STATEMENT.                    READ3710
          1770  NCARD  =  0                                                       READ3720
          1780  IF (NVALUE .LE. 0)  GO TO 9999                                   READ3730
370             NSTATN  =  NSTATN + 1                                             READ3740
                IF (NSTATN .GT. MNSTATE)  GO TO 1810                              READ3750
                KSNUM (1, NSTATN)  =  NVALUE                                      READ3760
                KSNUM (2, NSTATN)  =  9999                                        READ3770
                I9999  =  1                                                       READ3780
375             GO TO 9999                                                        READ3790
              C                                                                   READ3800
          1790  IF (LCHARS .GT. 3)  GO TO 1630                                   READ3810
                ICOUNT (1, 1)  =  ICOUNT (1, 1) - 1                             READ3820
                GO TO 9999                                                        READ3830
380           C                                                                   READ3840
          1800  PRINT   50, IERROR, (LSTATE(I), I=1, LCHARS)                     READ3850
                IF (ITYPE .NE. 45)  GO TO 1630                                    READ3860
                GO TO 1750                                                        READ3870
          1810  PRINT   60, MNSTATE                                             READ3880
385             PRINT   70, (LSTATE(I), I=1, LCHARS)                              READ3890
```

```
      C     DUMP THE REMAINDER OF THIS ROUTINE                                READ3900
       1820  PRINT   80, LCARD, NAME                                          READ3910
             REWIND LUSTATE                                                   READ3920
             IP2  =  72                                                       READ3930
390          N  =  7                                                          READ3940
      C     CHECK FOR AN END STATEMENT                                        READ3950
       1830  IF (ICHECK(END,3,N,IP2,LCARD(1),IP3))  GO TO 1850                READ3960
      C     CHECK FOR A DOLLAR, $, SIGN INDICATING A MULTIPLE STATEMENT.      READ3970
       1840  N  =  ISCANL (IPUNCT(6), N + 1, 72, LCARD(1)) + 1                READ3980
395          IF (N .LE. 72)  GO TO 1830                                       READ3990
      C     READ THE NEXT RECORD.                                             READ4000
             READ (LUIN,   10)  LCARD                                         READ4010
             ICOUNT (1, 1)  =  ICOUNT (1, 1) + 1                              READ4020
             IF (EOF(LUIN))   1860, 1820                                      READ4030
400   C     END FOUND, RESET AND START THE NEXT ROUTINE.                      READ4040
       1850  CALL RESETS                                                      READ4050
             GO TO 1000                                                       READ4060
      C     EOF, TERMINATE                                                    READ4070
       1860  IEOF  =  1                                                       READ4080
405          ICOUNT (1, 1)  =  ICOUNT (1, 1) - 1                              READ4090
       9999  RETURN                                                           READ4100
      C                                                                       READ4110
         10  FORMAT  ( 100A1 )                                               READ4120
         20  FORMAT  (  *0ERROR IN THE FIRST 5 COLUMNS OF THE RECORD * 80A1 / READ4130
410        1   * THIS RECORD HAS BEEN LEFT IN THE FINAL ROUTINE AS A COMMENT* READ4140
           2   )                                                              READ4150
         30  FORMAT  (  *0NO PROGRAM, SUBROUTINE, FUNCTION, OR BLOCK DATA  ST*READ4160
           1   *ATEMENT FOUND ON THE INPUT FILE FOR THIS ROUTINE.* /          READ4170
           2   * CHECK THE FIRST AND LAST TWO RECORDS OF THIS ROUTINE BEFORE *READ4180
415        3   *COMPILATION.*  )                                              READ4190
         40  FORMAT  (  *0COULD NOT FIND AN EQUAL SIGN IS THIS REPLACEMENT ST*READ4200
           1   *ATEMENT.* / (1X, 130A1) )                                     READ4210
         50  FORMAT  (  *0ERROR IN THE FOLLOWING STATEMENT. ITYPE = * IS /    READ4220
           1   (20X, 100A1) )                                                READ4230
420      60  FORMAT  (  *0THE ARRAY (KSNUM) IS FULL.  THE NUMBER OF EXECUTABL*READ4240
           1   *E STATEMENT NUMBERS EXCEEDED * I5 )                          READ4250
         70  FORMAT  (  *0THE PREVIOUS ERROR FORCED THE TERMINATION OF PROCES*READ4260
           1   *SING OF THE INPUT FOR THIS ROUTINE ON STATEMENT* / (20X,      READ4270
           2   100A1) )                                                      READ4280
425      80  FORMAT  (  * THIS INPUT RECORD NOT PROCESSED * 80A1 *  FOR ROUTI*READ4290
           1   *NE * 4A1 )                                                   READ4300
      C                                                                       READ4310
             END                                                              READ4320
```

```
      SUBROUTINE  RESETS                                                    RESE  10
C     THIS ROUTINE RESETS THE POINTERS AND COUNTERS.                        RESE  20
      COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,        RESE  30
     1   IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM    RESE  40
     2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),        RESE  50
     3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,     RESE  60
     4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING    RESE  70
     5   (2, 100)                                                          RESE  80
      COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT        RESE  90
     1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,        RESE 100
     2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,        RESE 110
     3   STAR, X                                                           RESE 120
      INTEGER     STRING                                                    RESE 130
        DO 1000  I  =  1, 7                                                 RESE 140
1000    NUMBER (I)  =  0                                                    RESE 150
        DO 1010  J  =  1, 100                                               RESE 160
          DO 1010  I  =  1, 2                                               RESE 170
1010      STRING (I, J)  =  IBLANK                                          RESE 180
        DO 1020  I  =  1, 4                                                 RESE 190
1020    NAME (I)  =  IBLANK                                                 RESE 200
        DO 1030  I  =  1, 1000                                              RESE 210
        LFOUT (I)  =  IBLANK                                                RESE 220
1030    LSTATE (I)  =  IBLANK                                               RESE 230
        DO 1040  J  =  1, 100                                               RESE 240
        KFORM (J)  =  0                                                     RESE 250
          DO 1040  I  =  1, 3                                               RESE 260
1040      KFOUT (I, J)  =  0                                                RESE 270
        DO 1050  I  =  1, 4                                                 RESE 280
1050    ICOUNT (I, 1)  =  0                                                 RESE 290
      ICHARS  =  0                                                          RESE 300
      IERROR  =  0                                                          RESE 310
      IPROG  =  0                                                           RESE 320
      I9999  =  0                                                           RESE 330
      LCHARS  =  0                                                          RESE 340
      NCARDS  =  0                                                          RESE 350
      NEXT  =  1                                                            RESE 360
      NFORMN  =  0                                                          RESE 370
      NFOUT  =  0                                                           RESE 380
      NKFORM  =  0                                                          RESE 390
      NOUTS  =  0                                                           RESE 400
      NSNUMC  =  990                                                        RESE 410
      NSTATN  =  0                                                          RESE 420
      NUMK  =  0                                                            RESE 430
      ENTRY RESETX                                                          RESE 440
        DO 1060  J  =  1, NUMMAX                                            RESE 450
          DO 1060  I  =  1, 2                                               RESE 460
1060      INNUM (I, J)  =  0                                                RESE 470
      ISNUM  =  0                                                           RESE 480
      ITYPE  =  1                                                           RESE 490
      NVALUE  =  0                                                          RESE 500
      NUMIN  =  0                                                           RESE 510
9999  RETURN                                                                RESE 520
      END                                                                   RESE 530
```

SUBROUTINE SHIFTR CDC 6600 FTN V3.0-P355 OPT=1 06/25/75 12.55.39.

```
      SUBROUTINE  SHIFTR (NEW, ISTART, ISTOP, LIST)                    SHIF  10
C     THIS ROUTINE SHIFTS ALL DATA IN THE LIST FROM ISTART THRU ISTOP   SHIF  20
C       ONE SPACE TO THE RIGHT.  THE CREATED SPACE IS FILLED BY NEW.   SHIF  30
        DIMENSION   LIST (1)                                           SHIF  40
 5      I  =  ISTOP                                                    SHIF  50
 1000   LIST (I + 1)  =  LIST (I)                                      SHIF  60
        I  =  I - 1                                                    SHIF  70
        IF (I .GE. ISTART)  GO TO 1000                                 SHIF  80
        LIST (ISTART)  =  NEW                                          SHIF  90
10      ISTOP  =  ISTOP + 1                                            SHIF 100
        GO TO 9999                                                     SHIF 110
        ENTRY SHIFTL                                                   SHIF 120
C     THIS ROUTINE SHIFTS ALL DATA IN THE LIST FROM ISTART THRU ISTOP   SHIF 130
C       ONE SPACE TO THE LEFT.   THE CREATED SPACE IS FILLED BY NEW.   SHIF 140
15    C     NOTICE...  THE VALUE OF ISTOP IS ADJUSTED.                 SHIF 150
        ISTOP  =  ISTOP - 1                                            SHIF 160
        IF (ISTART .GT. ISTOP)  GO TO 1020                             SHIF 170
          DO 1010  I  =  ISTART, ISTOP                                 SHIF 180
 1010     LIST (I)  =  LIST (I + 1)                                     SHIF 190
20      1020  LIST (ISTOP + 1)  =  NEW                                 SHIF 200
        9999  RETURN                                                   SHIF 210
        END                                                            SHIF 220
```

SUBROUTINE SPACOUT CDC 6600 FTN V3.0-P355 OPT=1 06/25/75 12.55.39.

```
      SUBROUTINE  SPACOUT                                                SPAC  10
C     THIS ROUTINE INSERTS THE COMMON SPACINGS.                          SPAC  20
      COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,     SPAC  30
     1   IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM  SPAC  40
     2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),      SPAC  50
     3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,   SPAC  60
     4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING  SPAC  70
     5   (2, 100)                                                       SPAC  80
      COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT      SPAC  90
     1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,      SPAC 100
     2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,      SPAC 110
     3   STAR, X                                                        SPAC 120
      DIMENSION   LIST (1)                                              SPAC 130
      INTEGER     H                                                     SPAC 140
      EQUIVALENCE  (LIST(1), LSTATE(1))                                 SPAC 150
      II  =  IPOINT                                                     SPAC 160
C     I4    POSITION OF NEXT ).                                         SPAC 170
      I4  =   - 1000                                                    SPAC 180
 1000 IFLAG  =  0                                                       SPAC 190
 1010 IF (II .GT. ICHARS)  GO TO 9999                                   SPAC 200
        DO 1020  J  =  1, I0                                            SPAC 210
        IF (LIST(II) .EQ. INTEGER(J))  GO TO 1050                       SPAC 220
 1020   CONTINUE                                                        SPAC 230
 1030 IF (LIST(II) .EQ. IPUNCT(1))  GO TO 1090                          SPAC 240
      IF (LIST(II) .EQ. IPUNCT(2))  GO TO 1110                          SPAC 250
      IF (LIST(II) .EQ. IPUNCT(3))  GO TO 1120                          SPAC 260
      IF (LIST(II) .EQ. IPUNCT(5))  GO TO 1090                          SPAC 270
      IF (LIST(II) .EQ. IPUNCT(9))  GO TO 1090                          SPAC 280
      IF (LIST(II) .EQ. IPUNCT(10))  GO TO 1090                         SPAC 290
 1040 II  =  II + 1                                                     SPAC 300
      GO TO 1000                                                        SPAC 310
C                                                                       SPAC 320
 1050 N  =  J - 1                                                       SPAC 330
 1060 II  =  II + 1                                                     SPAC 340
      IF (II .GT. ICHARS)  GO TO 9999                                   SPAC 350
      IF (LIST(II) .EQ. IBLANK)  GO TO 1060                             SPAC 360
        DO 1070  J  =  1, 10                                            SPAC 370
        IF (LIST(II) .EQ. INTEGER(J))  GO TO 1080                       SPAC 380
 1070   CONTINUE                                                        SPAC 390
      IF (LIST(II) .NE. H)  GO TO 1030                                  SPAC 400
      II  =  II + 1 + N                                                 SPAC 410
      GO TO 1000                                                        SPAC 420
 1080 N  =  J - 1 + N * 10                                              SPAC 430
      GO TO 1060                                                        SPAC 440
 1090 CALL INSERT (IBLANK, II + 1, LCHARS, LIST(1), 1)                  SPAC 450
      CALL INSERT (IBLANK, II, LCHARS, LIST(1), 1)                      SPAC 460
      II  =  II + 3                                                     SPAC 470
      I4  =  I4 + 2                                                     SPAC 480
 1100 IFLAG  =  1                                                       SPAC 490
      GO TO 1010                                                        SPAC 500
C    INSERT AFTER ,                                                     SPAC 510
 1110 CALL INSERT (IBLANK, II + 1, LCHARS, LIST(1), 1)                  SPAC 520
      II  =  II + 2                                                     SPAC 530
      I4  =  I4 + 1                                                     SPAC 540
      GO TO 1100                                                        SPAC 550
 1120 IF (I4 .GT. 0)  GO TO 1130                                        SPAC 560
      I4  =  MATCH (II, ICHARS, LIST(1))                                SPAC 570
      GO TO 1140                                                        SPAC 580
 1130 IF (II .LT. I4)  GO TO 1040                                       SPAC 590
      I4  =   - 1000                                                    SPAC 600
C    INSERT BEFORE (                                                    SPAC 610
 1140 IF (IFLAG .EQ. 1)  GO TO 1040                                     SPAC 620
      CALL INSERT (IBLANK, II, LCHARS, LIST(1), 1)                      SPAC 630
      II  =  II + 2                                                     SPAC 640
      I4  =  I4 + 1                                                     SPAC 650
      GO TO 1000                                                        SPAC 660
 9999 RETURN                                                            SPAC 670
      END                                                               SPAC 680
```

```
 FUNCTION    SPRESS                                CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

                FUNCTION  SPRESS (I, ISTOP, LIST)                                  SPRE  10
              C   THIS ROUTINE SURPRESSES ALL BLANKS.                              SPRE  20
                DIMENSION   LIST (1)                                               SPRE  30
                DATA    IB / 1H   /                                                SPRE  40
 5              SPRESS  =  0.0                                                     SPRE  50
               1000  IF (I .GT. ISTOP)  GO TO 1010                                SPRE  60
                IF (LIST(I) .NE. IB)  GO TO 9999                                   SPRE  70
              C    SURPRESS ANY STRAY BLANKS.                                      SPRE  80
                CALL SHIFTL (IB, I, ISTOP, LIST(1))                                SPRE  90
10              GO TO 1000                                                         SPRE 100
               1010  SPRESS  =  1.0                                               SPRE 110
               9999  RETURN                                                       SPRE 120
                END                                                                SPRE 130
```

```
      SUBROUTINE  STORE (JTYPE)                                          STOR  10
C     THIS ROUTINE  ADDS DIMENSION AND TYPED VARIABLES OF TYPE JTYP TO   STOR  20
C       THE ARRAY STRING.                                                STOR  30
        COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,   STOR  40
     I    IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM STOR  50
     2    (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),     STOR  60
     3    LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  STOR  70
     4    NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING STOR  80
     5    (2, 100)                                                       STOR  90
      COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT      STOR 100
     1    (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,     STOR 110
     2    MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,     STOR 120
     3    STAR, X                                                        STOR 130
      DIMENSION    ITESTN ( 7), ITESTN1 ( 7), LIST (1), NEWORD (2)       STOR 140
      INTEGER      STRING                                                STOR 150
      EQUIVALENCE  (II, IPOINT), (ISTOP, ICHARS), (LIST(1), LSTATE(1))   STOR 160
C     RANGE OF THE LOCATIONS NI THRU N.                                  STOR 180
        N  =  0                                                          STOR 190
          DO 1000  I  =  I, JTYPE                                        STOR 200
 1000     N  =  N + NUMBER (I)                                           STOR 210
        N1  =  N - NUMBER (JTYPE) + 1                                    STOR 220
 1010   I3  =  ISCANL (IPUNCT(3), II, ISTOP, LIST(1))                    STOR 230
        IS  =  ISCANL (IPUNCT(2), II, ISTOP, LIST(I)) - 1                STOR 240
        IF (I3-IS-I)   1020, 1030, 1040                                  STOR 250
 1020   CALL INSERT (IBLANK, I3, LCHARS, LIST(I), 1)                     STOR 260
        IS  =  MATCH (I3 + I, ISTOP, LIST(1))                            STOR 270
        GO TO 1040                                                       STOR 280
 1030   IS  =  ISTOP                                                     STOR 290
 1040   LENGTH  =  MIN0 (20, IS - II + 1)                                STOR 300
        IF (LENGTH .LE. 0)  GO TO 9999                                   STOR 310
        NEWORD (I)         =  IBLANK                                     STOR 320
        NEWORD (2)         =  IBLANK                                     STOR 330
        ENCODE (LENGTH,    10, NEWORD (1))          (LIST (K), K=11, IS) STOR 340
        DECODE ( 7,    10, NEWORD (I))         ITESTN                    STOR 350
        IF (N .LT. N1)  GO TO 1060                                       STOR 360
C     CHECK IF THIS VARIABLE IS ALREADY PRESENT IN THE STRING,           STOR 370
C       IF SO DROP IT.                                                   STOR 380
          DO 1050  J  =  N1, N                                           STOR 390
          IF (NEWORD(1)       .EQ. STRING(1,J))  GO TO 1150              STOR 400
 1050     CONTINUE                                                       STOR 410
C     PUSH DOWN THE STRING.                                              STOR 420
 1060   K  =  NUMK  =  NUMK + 1                                          STOR 430
        IF (NUMK .LE. NMAX)  GO TO 1070                                  STOR 440
        PRINT   20, NEWORD                                               STOR 450
        GO TO 9999                                                       STOR 460
 1070   IF (K .LE. N1)  GO TO 1090                                       STOR 470
          DO 1080  1  =  I, 2                                            STOR 480
 1080     STRING (I, K)  =  STRING (I, K - 1)                            STOR 490
        K  =  K - 1                                                      STOR 500
        IF (K .GT. N)  GO TO 1070                                        STOR 510
C     INSERT THE NEW VARIABLE DEFINITION.                                STOR 520
 1090   NN  =  N  =  N + 1                                               STOR 530
        NUMBER (JTYPE)  =  NUMBER (JTYPE) + I                            STOR 540
          DO 1100  1  =  1, 2                                            STOR 550
 1100     STRING (1, N)  =  NEWORD (I)                                   STOR 560
```

```
  SUBROUTINE  STORE                                CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

             1110  IF (NN .LE. N1)  GO TO 1150                                          STOR 570
                   NN  =  NN - 1                                                       STOR 580
                   DECODE ( 7,   10, STRING (1, NN))  ITESTN1                          STOR 590
                   LENGTH  =  MIN0 (LENGTH, 7)                                         STOR 600
60                   DO 1120  I  =  1, LENGTH                                         STOR 610
                     IF (ITESTN1(I) .EQ. IBLANK)  GO TO 1150                          STOR 620
                     IF (ITESTN(I) .EQ. IBLANK)  GO TO 1130                           STOR 630
                     IF (ITESTN(I)  -   ITESTN1(I))  1130, 1120, 1150                 STOR 640
             1120    CONTINUE                                                         STOR 660
65           1130    DO 1140  I  =  1, 2                                              STOR 670
                     STRING (I, NN + 1)  =  STRING (I, NN)                            STOR 680
             1140    STRING (I, NN)  =  NEWORD (I)                                    STOR 690
                   GO TO 1110                                                         STOR 700
             1150  II  =  IS + 2                                                      STOR 710
70                 IF (II .LE. ISTOP)  GO TO 1010                                   STOR 720
             9999  RETURN                                                           STOR 730
            C                                                                       STOR 740
               10  FORMAT  ( 100A1 )                                                STOR 750
               20  FORMAT  (  *0THE ARRAY STRING IS FULL, THE VARIABLES STARTING WI*STOR 760
75               1   *TH * 2A10 * WERE DROPPED.*  )                                 STOR 770
            C                                                                       STOR 780
                   END                                                              STOR 790
```

SUBROUTINE SUMMARY CDC 6600 FTN V3.0-P355 OPT=1 06/25/75 12.55.39.

```
      SUBROUTINE  SUMMARY                                                    SUMM  10
C     THIS ROUTINE PRODUCES THE SUMMARY REPORT AFTER EACH ROUTINE HAS        SUMM  20
C       BEEN PROCESSED.                                                      SUMM  30
      COMMON  /ALL/  ICHARS, 1DOLLAR, 1ERROR, INNUM (2, 50), IPOINT,         SUMM  40
     1   IPROG, 1SNUM, 1TYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM     SUMM  50
     2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),          SUMM  60
     3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,       SUMM  70
     4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING      SUMM  80
     5   (2, 100)                                                            SUMM  90
      COMMON  /DATA/  C, END, H, 1BLANK, IEOF, INTEGER (10), IPUNCT          SUMM 100
     1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,          SUMM 110
     2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,          SUMM 120
     3   STAR, X                                                             SUMM 130
      DIMENSION   PER (2,2)                                                  SUMM 140
      INTEGER     PROGRAM                                                    SUMM 150
        DO 1000  J  =  1, 3                                                  SUMM 160
 1000   ICOUNT (2, J)  =  ICOUNT (2, J) + ICOUNT (1, J)                      SUMM 170
        DO 1010  I  =  1, 2                                                  SUMM 180
 1010   ICOUNT (I, 4)  =  ICOUNT (1, 2) - ICOUNT (I, 3)                      SUMM 190
        DO 1040  I  =  1, 2                                                  SUMM 200
          DO 1020  J  =  1, 2                                                SUMM 210
 1020     PER (I, J)  =  0.0                                                 SUMM 220
        IF (ICOUNT(I,2) .EQ. 0)  GO TO 1040                                  SUMM 230
          DO 1030  J  =  1, 2                                                SUMM 240
 1030     PER (I, J)  =  100. * ICOUNT (I, J + 2) / ICOUNT (I, 2)            SUMM 250
 1040   CONTINUE                                                            SUMM 260
      PRINT   10, NAME, PROGRAM, ((ICOUNT(I, J), I=1, 2), J=1, 2), (         SUMM 270
     I   (ICOUNT(1, J + 2), PER(I, J), 1=1, 2), J=1, 2)                      SUMM 280
      IF (NFORMN .LE. 0)  GO TO 1050                                         SUMM 290
      PRINT   20, (J, KFORM(J), J=1, NFORMN)                                 SUMM 300
      GO TO 1060                                                             SUMM 310
 1050 PRINT   30                                                             SUMM 320
 1060 IF (NSTATN .LE. 0)  GO TO 1070                                         SUMM 330
      PRINT   40, ((KSNUM(1, J), I=1, 2), J=1, NSTATN)                       SUMM 340
      GO TO 9999                                                             SUMM 350
 1070 PRINT   50                                                             SUMM 360
 9999 RETURN                                                                 SUMM 370
C                                                                            SUMM 380
   10 FORMAT  (  *0COUNTER SUMMARY* 44X,  *FOR ROUTINE * 4A1, 24X,           SUMM 390
     1   *CUMMULATIVE FOR PROGRAM * 7A1 // 36X, 2(30X,  *PERCENT    * ) /SUMM 400
     2   46X, 2(20X,  *OUTPUT                * ) /  *0NUMBER OF INPUT REC*SUMM 410
     3   *ORDS.                 * 2(I20, 20X) /  * NUMBER OF OUTPUT RECO*SUMM 420
     4   *RDS.                * 2(I20, 20X) /  * NUMBER OF COMMENT STATE*SUMM 430
     5   *MENTS.            * 2(I20, F10.1, 10X) /  * NUMBER OF VALID EX*SUMM 440
     6   *ECUTABLE STATEMENTS.  * 2(I20, F10.1, 10X) )                       SUMM 450
   20 FORMAT  (  *0FORMAT STATEMENT NUMBER INDEX              NEW/OLD*       SUMM 460
     1   2X, 6(I6 *0/* I5) / (10(I6 *0/* I5)) )                             SUMM 470
   30 FORMAT  (  *0THIS ROUTINE USES NO FORMAT STATEMENTS*  )                SUMM 480
   40 FORMAT  (  *0EXECUTABLE STATEMENT NUMBER INDEX          OLD/NEW*       SUMM 490
     1   2X, 6(I7 */* I5) / (10(I7 */* I5)) )                               SUMM 500
   50 FORMAT  (  *0THIS ROUTINE USES NO EXECUTABLE STATEMENT NUMBERS*        SUMM 510
     1   )                                                                   SUMM 520
C                                                                            SUMM 530
      END                                                                    SUMM 540
```

```
FUNCTION    TRANSF                          CDC 6600 FTN V3.0-P355 OPT=1   06/25/75  12.55.39.

      INTEGERFUNCTION  TRANSF (I1, I2)                                          TRAN  10
C     THIS ROUTINE TRANSFERS THE DATA RECORD FROM ICARD TO ISTATE.              TRAN  20
      COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,            TRAN  30
     1   IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM        TRAN  40
     2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),            TRAN  50
     3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,         TRAN  60
     4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING        TRAN  70
     5   (2, 100)                                                              TRAN  80
      COMMON  /DATA/  C, END, F, IBLANK, IEOF, INTEGER (10), IPUNCT             TRAN  90
     1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,            TRAN 100
     2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,            TRAN 110
     3   STAR, X                                                               TRAN 120
      TRANSF  =  0                                                             TRAN 130
        DO 1000  I  =  I1, I2                                                  TRAN 140
        IF (LCHARS .GE. MLCHARS)  GO TO 1010                                   TRAN 150
        LCHARS  =  LCHARS + 1                                                  TRAN 160
 1000   LSTATE (LCHARS)  =  LCARD (I)                                          TRAN 170
      GO TO 1020                                                               TRAN 180
 1010 PRINT   10, MLCHARS, LCARD                                               TRAN 200
      TRANSF  =  I                                                             TRAN 210
      LCHARS  =  MLCHARS                                                       TRAN 220
 1020 ICHARS  =  LCHARS                                                        TRAN 230
 9999 RETURN                                                                   TRAN 240
C                                                                              TRAN 250
   10 FORMAT  (  *0THE ARRAY (LSTATE) IS FULL, THE NUMBER OF CHARACTER*TRAN 260
     1   *S IN THE CURRENT STATEMENT EXCEEDED * I5 /  *0THE ARRAY LSTAT*TRAN 270
     2   *E OVERFLOWED ON CARD     * 80A1 )                                    TRAN 280
C                                                                              TRAN 290
      END                                                                      TRAN 300
```

SUBROUTINE WRITES CDC 6600 FTN V3.0-P355 OPT=1 06/25/75 12.55.39.

```
      SUBROUTINE  WRITES                                                WRIT  10
C     THIS ROUTINE CONTROLS THE WRITING OF THE OUTPUT FILE AND REPORT.  WRIT  20
      COMMON  /ALL/  ICHARS, IDOLLAR, IERROR, INNUM (2, 50), IPOINT,    WRIT  30
     1   IPROG, ISNUM, ITYPE, I9999, KFORM (100), KFOUT (3, 100), KSNUM WRIT  40
     2   (2, 400), LCARD (80), LCHARS, LFOUT (1000), LSTATE (2000),     WRIT  50
     3   LWORDS, NAME (4), NCARDS, NEXT, NFORMN, NFOUT, NKFORM, NOUTS,  WRIT  60
     4   NPUSH, NSNUMC, NSTATN, NUMBER (7), NUMIN, NUMK, NVALUE, STRING WRIT  70
     5   (2, 100)                                                       WRIT  80
      COMMON  /DATA/  C, END, H, IBLANK, IEOF, INTEGER (10), IPUNCT     WRIT  90
     1   (11), ICOUNT (2, 4), LUIN, LUOUT, LUSTATE, MFOUT, MLCHARS,     WRIT 100
     2   MNFORM, MNSTATE, NCARD, NMAX, NUMMAX, PROGRAM (7), RETURN,     WRIT 110
     3   STAR, X                                                        WRIT 120
      DIMENSION   KCARD (200), NPSTACK (10)                             WRIT 130
      INTEGER     C, END, H, RETURN, STAR, X                            WRIT 140
      IFLAG  =  0                                                       WRIT 150
      IN  =  0                                                         WRIT 160
      NOLDTYP  =  0                                                     WRIT 170
      NPUFLAG  =  0                                                     WRIT 180
      NPUSH  =  0                                                       WRIT 190
      REWIND LUSTATE                                                    WRIT 200
      IF (NOUTS .LE. 0)  GO TO 9999                                     WRIT 210
 1000 READ (LUSTATE)  NTYPE,LWORDS, IC, ISNUM, (KCARD(I), I=1, LWORDS),WRIT 220
     1   NUMIN, ( (INNUM(I, J), I=1, 2), J=1, NUMIN)                    WRIT 230
      IF (EOF(LUSTATE))   1220, 1010                                    WRIT 240
 1010 NOUTS  =  NOUTS - 1                                               WRIT 250
        DO 1020  I  =  1, 1000                                          WRIT 260
 1020   LSTATE (I)  =  IBLANK                                          WRIT 270
      IF (NTYPE .EQ. 0)  GO TO 1190                                     WRIT 280
      IFLAG  =  0                                                       WRIT 290
      IF (NTYPE .GE. 15 .AND. NOLDTYP .LE. 6)  CALL OUTSTR              WRIT 300
      NOLDTYP  =  NTYPE                                                 WRIT 310
      LCHARS  =  IC + 7                                                 WRIT 320
      IF (IC .GT. 100)  GO TO 1030                                      WRIT 330
      IF (IC .LE. 0)  GO TO 1000                                        WRIT 340
      DECODE (IC,   10, KCARD (1))   (LSTATE (I), I=8, LCHARS)          WRIT 350
      GO TO 1050                                                        WRIT 360
 1030 II  =  1                                                          WRIT 370
      I1  =  8                                                          WRIT 380
 1040 I2  =  MIN0 (I1 + 99, LCHARS)                                     WRIT 390
      ICC  =  MIN0 (IC, 100)                                            WRIT 400
      IF (ICC .LE. 0)  GO TO 1050                                       WRIT 410
      DECODE (ICC,   10, KCARD (II))   (LSTATE (I), I=I1, I2)           WRIT 420
      IC  =  IC - 100                                                   WRIT 430
      II  =  10 + II                                                    WRIT 440
      I1  =  100 + I1                                                   WRIT 450
      IF (IC .GT. 0)  GO TO 1040                                        WRIT 460
 1050 IF (NTYPE .NE. 18)  GO TO 1060                                    WRIT 470
C     RECORD THE DO LOOP TERMINAL POINT STATEMENT NUMBER.               WRIT 480
      NPUSH  =  NPUSH + 1                                               WRIT 490
      NPSTACK (NPUSH)  =  INNUM (2, 1)                                  WRIT 500
 1060 IF (ISNUM .EQ. 0)  GO TO 1120                                     WRIT 510
C     LABEL THE NEW STATEMENT NUMBER                                    WRIT 520
      ENCODE (5,   20, L)  ISNUM                                        WRIT 530
      DECODE (5,   10, L)   (LSTATE(I), I=1, 5)                         WRIT 540
      IF (NPUSH .EQ. 0)  GO TO 1120                                     WRIT 550
```

```
   SUBROUTINE  WRITES                                CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

            C    CHECK FOR THE DO LOOP TERMINATION STATEMENT NUMBER.                                WRIT 560
                   NPU  =  NPUSH                                                                    WRIT 570
                     DO 1070  J  =  1, NSTATN                                                       WRIT 580
                     IF (KSNUM(2,J) .EQ. ISNUM)  GO TO 1080                                         WRIT 590
60           1070    CONTINUE                                                                       WRIT 600
                   GO TO 1120                                                                       WRIT 610
             1080    DO 1090  I  =  1, NPU                                                          WRIT 620
                     IF (NPSTACK(I) .EQ. KSNUM(1,J))  GO TO 1100                                    WRIT 630
             1090    CONTINUE                                                                       WRIT 640
65                 GO TO 1120                                                                       WRIT 650
            C    IF THIS IS A TERMINATION STATEMENT REDUCE THE PUSH COUNT AND THE STWRIT 660
             1100  NPUFLAG  =  NPUFLAG + 1                                                         WRIT 670
                   NPU  =  NPUSH - NPUFLAG                                                          WRIT 680
                     DO 1110  II  =  I, NPU                                                         WRIT 690
70           1110    NPSTACK (II)  =  NPSTACK (II + 1)                                            WRIT 700
                   IF (NPU .GE. I)  GO TO 1080                                                      WRIT 710
             1120  IF (NUMIN .LE. 0)  GO TO 1180                                                  WRIT 720
            C    INSERT ALL REVISED INTERNAL STATEMENT NUMBERS                                      WRIT 730
                     DO 1130  J  =  1, NSTATN                                                       WRIT 740
75                   IF (INNUM(2,NUMIN) .EQ. KSNUM(1,J))  GO TO 1150                              WRIT 750
             1130    CONTINUE                                                                       WRIT 760
                     DO 1140  J  =  1, NFORMN                                                       WRIT 770
                     IF (INNUM(2,NUMIN) .EQ. KFORM(J))  GO TO 1160                                  WRIT 780
             1140    CONTINUE                                                                       WRIT 790
80                 PRINT   30, INNUM (2, NUMIN)                                                     WRIT 800
                   CALL INSERTN (INNUM(2, NUMIN), INNUM(1, NUMIN) + 7, LCHARS,                      WRIT 810
                  1   LSTATE(1), 0)                                                                 WRIT 820
                   GO TO 1170                                                                       WRIT 830
             1150  CALL INSERTN (KSNUM(2, J), INNUM(1, NUMIN) + 7, LCHARS,                          WRIT 840
85                1   LSTATE(1), 0)                                                                WRIT 850
                   GO TO 1170                                                                       WRIT 860
             1160  CALL INSERTN (J * 10, INNUM(1, NUMIN) + 7, LCHARS, LSTATE(1), 4)                 WRIT 870
             1170  NUMIN  =  NUMIN - 1                                                             WRIT 880
                   GO TO 1120                                                                       WRIT 890
90          C                                                                                       WRIT 900
             1180  IF (NPUSH .LE. 0)  GO TO 1200                                                   WRIT 910
            C    PUSH OVER THE STATEMENT AS REQUIRED.                                               WRIT 920
                   CALL INSERT (IBLANK, 8, LCHARS, LSTATE(1), 2 * NPUSH)                            WRIT 930
                   GO TO 1200                                                                       WRIT 940
95          C    PROCESS A COMMENT STATEMENT.                                                       WRIT 950
             1190  IC  =  MIN0 (IC, 72)                                                            WRIT 960
                   LCHARS  =  IC                                                                    WRIT 970
                   ICOUNT (1, 3)  =  ICOUNT (1, 3) + 1                                              WRIT 980
            C    SKIP DOUBLE BLANK RECORDS IN SUCCESSION.                                           WRIT 990
100                IF (IFLAG .EQ. 1 .AND. IC .LE. 1)  GO TO 1210                                   WRIT1000
                   IFLAG  =  0                                                                      WRIT1010
                   IF  (IC .LE. 1)  IFLAG  =  1                                                     WRIT1020
                   DECODE (IC,   10, KCARD (1))   (LSTATE (I), I=1, IC)                             WRIT1030
            C                                                                                       WRIT1040
105          1200  IF (NTYPE .EQ. 36 .AND. NOUTS .EQ. 0)  GO TO 1230                              WRIT1050
                   IF (NTYPE .EQ. 21)  CALL IFSPACE                                                 WRIT1060
                   CALL PUNCHIT (NTYPE)                                                             WRIT1070
             1210  NPUSH  =  NPUSH - NPUFLAG                                                       WRIT1080
                   NPUFLAG  =  0                                                                    WRIT1090
110                IF (NOUTS .GT. 0)  GO TO 1000                                                   WRIT1100
```

```
   SUBROUTINE  WRITES                                CDC 6600 FTN V3.0-P355 OPT=1  06/25/75  12.55.39.

             1220  IF (IPROG .GE. 4 .OR. IPROG .LE. 1)  GO TO 1280                                WRIT1110
                   IF (NTYPE .EQ. 20 .AND. I9999 .EQ. 0)  GO TO 1280                              WRIT1120
                   GO TO 1250                                                                     WRIT1130
             1230    DO 1240  I  =  1, LCHARS                                                     WRIT1140
115          1240    LSTATE (I)  =  IBLANK                                                        WRIT1150
                   ICHARS  =  LCHARS  =  0                                                        WRIT1160
             1250  CALL INSERT (RETURN, 1, LCHARS, LSTATE(1), 8)                                  WRIT1170
                   IF (I9999 .EQ. 0)  GO TO 1260                                                  WRIT1180
                   CALL INSERT (IBLANK, 1, LCHARS, LSTATE(1), 2)                                  WRIT1190
120                CALL INSERTN (9999, 1, LCHARS, LSTATE(1), 4)                                   WRIT1200
                   GO TO 1270                                                                     WRIT1210
             1260  CALL INSERT (IBLANK, 1, LCHARS, LSTATE(1), 7)                                  WRIT1220
             1270  CALL PUNCHIT (99)                                                              WRIT1230
             1280  CALL OUTFRM                                                                    WRIT1240
125                CALL INSERT (END, 1, LCHARS, LSTATE(1), 3)                                     WRIT1250
                   CALL INSERT (IBLANK, 1, LCHARS, LSTATE(1), 7)                                  WRIT1260
                   CALL PUNCHIT (100)                                                             WRIT1270
                   REWIND LUSTATE                                                                 WRIT1280
                   ICOUNT (1, 2)  =  ICOUNT (1, 2) + NCARDS                                       WRIT1290
130                CALL SUMMARY                                                                   WRIT1300
             9999  RETURN                                                                         WRIT1310
            C                                                                                     WRIT1320
               10  FORMAT  ( 100A1 )                                                             WRIT1330
               20  FORMAT  ( I5 )                                                                WRIT1340
135            30  FORMAT  (  *0STATEMENT NUMBER * I6 * WAS USED INTERNALLY IN A BU*WRIT1350
                 1   *T IT WAS NOT USED AS STATEMENT LABEL.  THE ORIGINAL VALUE WAS*WRIT1360
                 2   * REINSERTED. *  )                                                           WRIT1370
            C                                                                                     WRIT1380
                   END                                                                            WRIT1390
```

```
**************************************************************************
* IMPORTANT NOTICE...AUDITRS WILL ABORT IF THE CL PARAMETER ON YOUR      *
*  JOB CARD IS LESS THAN 41000.                                          *
**************************************************************************
      NOTICE TO ALL USERS   ....................                       *
*         C O B O L
*         *********
*         COBOL HAS BEEN BACKED UP TO LEVEL 336
*         ANY USER WHO HAD PROBLEMS WITH COBOL LEVEL 365
*         PLEASE CALL EXT. 4784
**************************************************************************
* NOTICE...THE RUN COMPILER HAS BEEN UPDATED TO PSR LEVEL 380.           *
**************************************************************************
  06/25/75  M.I.P.C. SERIAL 121 SCOPE 3.3 L355.126
12.54.36.CKGHOHA
12.54.36.IP  000001 INPUT UNITS USED.
12.54.36.$SEQUENCE,KGH.
12.54.36.
12.54.36.$CHARGE,T1308  -060.
12.54.36.
12.54.36.GETUM,CM12000,MT1,P4,T30,CL55000.
12.54.36.
12.54.36.LABEL,TAPE1,R,L=USBMSEPPANEN,VSN=X1851.
12.54.38.MT 50 ASSIGNED TO TAPE1
12.55.29. MT 50 VISUAL REEL NUMBER IS  0X1851
12.55.29. LABEL READ WAS
12.55.29.USBMSEPPANEN
12.55.29.   EDITION NUMBER    01
12.55.29.   RETENTION CYCLE   000
12.55.29.   CREATION DATE     75168
12.55.29.   REEL NUMBER       0001
12.55.29.SKIPF,TAPE1,8,17,B.
12.55.35.COPYBF(TAPE1,B)
12.55.36.FILE OPENED - B
12.55.38.REWIND,B.
12.55.38.RFL,55000.
12.55.39.CM  012000     CM CELLS USED.
12.55.39.CP  000000.054 CP SEC. USED.
12.55.39.IO  000011.119 IO SEC. USED.
12.55.39.SS  000000.466 SYSTEM SEC. USED.
12.55.39.FTN,I=B,R=0.
12.55.39.FILE OPENED - COMPS
12.55.39.FILE OPENED - OUTPUT
12.55.39.FILE OPENED - FTNRLST
12.55.39.FILE OPENED - LGO
12.56.36.IP  000001 STORAGE DATA BLOCKS ON FILE  INPUT
12.56.36.OP  000167 STORAGE DATA BLOCKS ON FILE  OUTPUT
12.56.36.CM  055000     CM CELLS USED.
12.56.36.CP  000021.880 CP SEC. USED.
12.56.36.IO  000047.522 IO SEC. USED.
12.56.36.SS  000019.703 SYSTEM SEC. USED.
12.56.36.AC -  END OF JOB.
```

APPENDIX B.--FUNCTION AND SUBROUTINE DESCRIPTIONS

Routine	Description
REOR	The main control routine. Establishes common values and controls the operation cycles. Calls RESETS, READS, and WRITES for each Fortran routine. Terminates with a call to EXIT after an EOF has been encountered.
BLANKS	A subroutine used to suppress blank spaces in the statement text. It offers special handling to the Hollerith fields in DATA and FORMAT statements. Uses following routines: INSERT, INSERTN, NONR, and SHIFTL.
CHECK (LOOK4, NN, ISTART, ISTOP, LIST, IPOINT)	A logical function that indicates whether the character string LOOK4, of length NN characters, was found in array LIST between the columns ISTART and ISTOP. Spaces in the LIST are suppressed. The position of the next character beyond the string identified is returned as IPOINT. Uses routine SHIFTL.
FIXDATA	A subroutine used to assure that the proper spacing is retained in DATA statement Hollerith fields.
IDENT (N)	A function that identifies the statement type. N indicates whether to look for a routine identification statement (N = 1) or subsequent statement (N = 2). Uses routine SHIFTL.
IFSPACE	A subroutine used to insert spacing in IF statements.
INSERT (NEW, ISTART, ISTOP, LIST, N)	A subroutine used to insert the character string NEW, of length N characters, into the array LIST immediately prior to ISTART. ISTOP indicates the upper range limit for LIST that must be shifted to make room for the new characters. It adjusts the statement number array INNUM to compensate for the inserted characters. Uses routine SHIFTR.
INSERTN	An entry in subroutine INSERT that inserts into LIST the character equivalent of the integer NEW. N is assumed to be 5.
INSERTS	An entry in subroutine INSERT that does the identical processing less the statement number readjustment in array INNUM.
ISCANL	An entry in function ISCANR that does the identical processing but starts the search at the left point. If the character is not found, the right point value plus 1 is returned.

Routine	Description
ISCANR (LOOK4, ISTART, ISTOP, LIST)	A function that returns the location of the first character matching LOOK4 in array LIST between the left point, ISTART, and the right point, ISTOP. The search is started at the right point. If a matching character is not found, the left point value less 1 is returned.
KF (NSTN)	A subroutine that catalogs the FORMAT statement number NSTN in array KFORM.
KLIST (IP, NSTN)	A logical function that indicates whether the internal statement number NSTN from the position indicated by IP has been properly cataloged in array INNUM.
KO (NSTN)	A logical function that indicates whether the FORMAT statement number NSTN has been properly cataloged in array KFOUT along with the statement's storage position and length.
MATCH (ISTART, ISTOP, LIST)	A function that returns the location of the matching right parenthesis corresponding to the left parenthesis in array LIST location ISTART. If a matching right parenthesis is not found, the value ISTOP plus 1 is returned.
NONL	An entry in function NONR which does the identical processing, but starts the search at the left point. If a nonmatching character is not found, the right point value plus 1 is returned.
NONR (LOOK4, ISTART, ISTOP, LIST)	A function that returns the location of the first character not matching LOOK4 in array LIST between the left point, ISTART, and the right point, ISTOP. The search is started at the right point. If a nonmatching character is not found, the left point value less 1 is returned.
NUMBS (ISTART, ISTOP, LIST)	A function that returns the integer value of the number beginning in column ISTART of array LIST. If a number is found, its digits are suppressed and all text in array LIST through column ISTOP is shifted left. If no number is found, a zero is returned. Uses routine SHIFTL.
OUTFRM	A subroutine used to reconstruct the required FORMAT statements from the array LFOUT. It is driven by the list of original statement numbers found in array KFORM. Uses following routines: INSERTN, INSERTS, ISCANL, and PUNCHIT.
OUTPUT (LIST)	A subroutine used to write on the work file the character string contained in array LIST.

Routine	Description
OUTSTR	A subroutine used to reconstruct the type statements from array STRING. It is driven by the number of typed variables contained in array NUMBER. Uses following routines: INSERTS, NONR, and PUNCHIT.
PUNCHIT (ITY)	A subroutine used to form the final set of records corresponding to the statement type indicated by ITY. It offers special handling to the Hollerith fields in DATA and FORMAT statements. Uses following routines: FIXDATA and ISCANR.
READS	A subroutine used to read the original routine from TAPE 2, to classify statement types, and to write the working file TAPE 10. Uses following routines: BLANKS, CHECK, IDENT, INSERT, ISCANL, KF, KLIST, KO MATCH, NONL, NONR, NUMBS, OUTPUT, RESETS, RESETX, SHIFTL, SPACOUT, STORE, and TRANSF.
RESETS	A subroutine that resets the pointers, counters, and arrays before each Fortran routine is processed.
RESETX	An entry point in subroutine RESETS that resets the pointers, counters, and arrays before each Fortran statement is processed.
SHIFTL	An entry point in subroutine SHIFTR that does the identical processing, but removes one character from the array. All text in array LIST through column ISTOP is shifted left one column.
SHIFTR (NEW, ISTART, ISTOP, LIST)	A subroutine that inserts the single character NEW into the array LIST just prior to column ISTART. All text in array LIST through column ISTOP is shifted right one column.
SPACOUT	A subroutine that inserts the standard spacing into the remainder of the Fortran statement. Uses following routines: INSERT and MATCH.
SPRESS (I, ISTOP, LIST)	A function that suppresses a string of blanks starting at LIST(I). All text in array LIST through column ISTOP is shifted left. Zero is returned if LIST(I) was not blank; one is returned if LIST(I) was blank.
STORE (JTYPE)	A subroutine that adds a new list of variables of the type indicated by JTYPE to those already stored in array STRING. The variables are alphanumerically sorted within type. Uses following routines: INSERT, ISCANL, and MATCH.

Routine	Description
SUMMARY	A subroutine that cumulates and prints the summary statistics for each routine that has been reorganized.
TRANSF(I1, I2)	An integer function that returns the number of characters transferred from array LCARD to array LSTATE. The range of the transfer from array LCARD is I1 through I2.
WRITES	A subroutine used to write the reorganized routine on TAPE 4. Uses following routines: INSERT, INSERTN, OUTFRM, OUTSTR, PUNCHIT, and SUMMARY.

APPENDIX C.--VARIABLE DEFINITIONS

```
APPENDIX C - VARIABLE DEFINITIONS          A FORTRAN ROUTINE REORGANIZER


C     C               ALPHA CHARACTER C.
C     END             ALPHA WORD END.
C     F               ALPHA CHARACTER H.
C     IA (I,J)        FORTRAN STATEMENT CHARACTER DECODER STRING,
C                       I = 1 JUMP ADDRESS IF CURRENT CHARACTER EXCEEDS
C                             MATCH CHARACTER,
C                       I = 2 MATCH CHARACTER,
C                       I = 3 NEXT ACTION IF MATCH,
C                         IF < 0 CHECK NEXT CHARACTER, IF MATCH ITYPE =
C                                ABSOLUTE VALUE OF IA (3,J)
C                         IF = 0 CHECK NEXT CHARACTER AND CONTINUE,
C                         IF > 0 ITYPE = IA (3,J)
C     IBLANK          ALPHA WORD BLANK.
C     ICHARS          LENGTH OF CURRENT STATEMENT UP TO THE $ SIGN, ≤ LCHARS.
C     IDOLLAR         INDICATES THE POSITION OF THE END OF THE CURRENT
C                       STATEMENT WHEN A $ SEPARATOR HAS BEEN USED.
C     IEOF            INDICATES THE EOF INDICATOR HAS BEEN ENCOUNTERED.
C                       0 → NO OR 1 → YES.
C     IERROR          ERROR INDICATOR, CORRESPONDS TO ITYPE.
C     INNUM (I,J)     INTERNAL STATEMENT NUMBER CODES,
C                       I = 1 CHARACTER COUNT POSITION IN STATEMENT,
C                       I = 2 ORIGINAL STATEMENT NUMBER.
C     INTEGER (I)     STRING OF INTEGERS IN CHARACTER FORMAT.
C     IPOINT          NEXT POSITION AFTER CHECK WORD.
C     IPROG           ROUTINE TYPE, CORRESPONDS TO ITYPE.
C                       100 → ERROR, NO ROUTINE TYPE RECORD.
C     IPUNCT  (I)     STRING OF PUNCTUATION MARKS IN CHARACTER FORMAT.
C     ISNUM           REVISED STATEMENT NUMBER FOR THE CURRENT STATEMENT.
C     ITYPE           STATEMENT TYPE
C         ROUTINE STATEMENTS:
C              1  PROGRAM          2  SUBROUTINE       3  FUNCTION
C              4  BLOCK DATA     100 ERROR
C         TYPE STATEMENTS:
C              5  COMMON/          6  COMMON           7  DIMENSION
C              8  EXTERNAL         9  COMPLEX         10 DOUBLE PRECISION
C             11 INTEGER          12 LOGICAL          13 REAL
C             41 TYPE
C         DEFINITION STATEMENTS:
C             15 EQUIVALENCE      16 DATA             17 FORMAT
C         EXECUTABLE STATEMENTS:
C             18 DO               19 GO TO (          20 GO TO
C             21 IF               22 CALL             23 ASSIGN
C             24 CONTINUE         25 READ (           26 READ
C             27 PRINT            28 WRITE (          29 PUNCH
C             30 BUFFER IN        31 BUFFER OUT       32 DECODE
C             33 ENCODE           34 STOP             35 ENTRY
C             36 RETURN           37 USE              38 ENDFILE
C             39 REWIND           40 BACKSPACE        42 PAUSE
C             43 NAMELIST         44 END              45 (REPLACEMENT)
C     I2              POSITION OF NEXT COMMA.
C     I3              POSITION OF NEXT LEFT PARENTHESIS.
C     I4              POSITION OF NEXT RIGHT PARENTHESIS.
C     I9999           INDICATES IF A RETURN STATEMENT HAS BEEN PROCESSED,
C                       0 → NO OR 1 → YES.
C     KFORM (I)       ORIGINAL FORMAT NUMBER LIST BY ORDER OF USAGE.
C     KFOUT (I,J)     FORMAT STATEMENT STORAGE DATA,
C                       I = 1 ORIGINAL STATEMENT NUMBER,
C                       I = 2 STARTING POSITION IN ARRAY LFOUT,
C                       I = 3 LENGTH OF STATEMENT IN CHARACTERS.
C     KSNUM (I,J)     STATEMENT NUMBER DATA,
```

```
C                    I = 1 ORIGINAL STATEMENT NUMBER,
C                    I = 2 NEW STATEMENT NUMBER.
C     LCARD (I)      INPUT DATA CARD RECORD IN CHARACTER FORM.
C     LCHARS         LENGTH OF CURRENT STATEMENT IN CHARACTERS
C                      ≤ MLCHARS = 2000.
C     LFOUT (I)      FORMAT STATEMENT STORAGE ARRAY.
C     LSTATE (I)     CURRENT STATEMENT IN CHARACTER FORM.
C     LUIN           LOGICAL UNIT OF THE INPUT FILE, TAPE2.
C     LUOUT          LOGICAL UNIT OF THE OUTPUT FILE, TAPE4.
C     LUSTATE        LOGICAL UNIT OF THE WORKING FILE, TAPE10.
C     LWORDS         LENGTH OF CURRENT STATEMENT IN WORDS ≤ 200.
C     NAME (I)       PROGRAM NAME ON OUTPUT RECORDS, IN CHARACTER FORMAT.
C     NCARD          NUMBER OF RECORDS READ FOR THE CURRENT STATEMENT.
C     NCARDS         NUMBER OF RECORDS WRITTEN FOR THE CURRENT ROUTINE.
C     NEXT           POINTER FOR THE ARRAY LFOUT ≤ MFOUT = 1000.
C     NFORMN         NUMBER OF FORMAT STATEMENTS ≤ MNFORM = 99.
C     NFOUT          POINTER FOR ARRAY KFOUT.
C     NOUTS          NUMBER OF OUTPUT RECORDS ON THE FILE LUSTATE.
C     NPUSH          NUMBER OF SIMULTANEOUS DO LOOPS.
C     NSNUMC         CURRENT NEW STATEMENT NUMBER.  INITIALLY = 990.
C     NSTATN         NUMBER OF PROGRAM STATEMENT NUMBERS ≤ MNSTATE = 400.
C     NUMIN          NUMBER OF INTERNAL STATEMENT NUMBERS FOR CURRENT
C                      STATEMENT, POINTER IN ARRAY INNUM ≤ NUMMAX = 50.
C     NUMK           NUMBER OF WORD PAIRS IN ARRAY STRING ≤ NMAX = 100.
C     NUMBER (I)     NUMBER OF VARIABLES OF EACH TYPE IN (STRING).
C                      I = 1 DIMENSION          I = 2 EXTERNAL
C                      I = 3 COMPLEX            I = 4 DOUBLE PRECISION
C                      I = 5 INTEGER            I = 6 LOGICAL
C                      I = 7 REAL
C     NVALUE         ORIGINAL STATEMENT NUMBER OF THE CURRENT STATEMENT.
C     RETURN         ALPHA WORD RETURN.
C     STAR           ALPHA CHARACTER *.
C     STRING (I,J)   STORAGE ARRAY FOR TYPE STATEMENT VARIABLES.
C     X              ALPHA CHARACTER X.
```

APPENDIX D.--SCOPE CONTROL CARDS


```
ATTACH,REOR,REOR.
REQUEST,TAPE4,*PF.
ATTACH,TAPE2,SOURCEPROGRAM.
RFL(55000)
REDUCE.
SET(0)
MODE(0)
REOR.
CATALOG,TAPE4,SOURCEPROGRAMREOR.
```


INT.-BU.OF MINES,PGH.,PA. 20709

☆ U.S. GOVERNMENT PRINTING OFFICE: 1975-603-755/50